Operating Instructions

HDRC4 - camera

*LOGLUX*Ò

Kamera Werk Dresden GmbH

Operating Instructions *LOGLUX*®

Kamera Werk Dresden GmbH
Bismarckstr. 56
01257 Dresden

Last changes:

November 5, 1998
November 9, 1998
November 16, 1998
November 19, 1998
December 18, 1998

Table of Contents

1 Introduction

1.1 The binary logarithm

Definition of the binary logarithm:

$$a = 2^{(\mathrm{lb}(a))}$$

$$\mathrm{lb}(a) = \frac{\log_x(a)}{\log_x(2)} = \frac{\ln(a)}{\ln(2)}$$

For the octave skip *2a* is:

$$\mathrm{lb}(2a) = \mathrm{lb}(2) + \mathrm{lb}(a) = 1 + \mathrm{lb}(a)$$

1.2 The *Fixed Pattern Noise*

Every pixel of the HDRC4-sensor used with the coordinates *(x,y)* alters the irradiance $E_e(x,y)$ of the sensor surface into an electrical voltage $U_S(x,y)$:

$$U_S(x,y) = U_1(x,y) \cdot \mathrm{lb}\left(\frac{E_e(x,y)}{E_{e0}}\right) + U_0(x,y) \qquad E_{e0} = 1\frac{W}{m^2}$$

The voltages $U_0(x,y)$ and $U_1(x,y)$ are coordinate-depending and normally distributed quantities. The visual effect resulting from this is called „*fixed pattern noise*“ (short FPN). The term *noise* is only indirectly accurate, because it is a noise in the local area and not in the time area. (The voltage constants $U_0(x,y)$ and $U_1(x,y)$ are temporally constant quantities.)

1.3 Correction of the *Fixed Pattern Noise*

In order to receive an output signal not depending on the coordinates with the same irradiance of all pixels, the FPN has to be corrected when reading out the sensor. This correction is executed in an arithmetic processing unit. The voltage dependig on the irradiance is multiplied with a gain correction voltage $U_{1korr}(x,y)$ and added up with an offset correction voltage $U_{0korr}(x,y)$:

$$U_{Skorr}(x,y) = U_S(x,y) \cdot U_{1korr}(x,y) + U_{0korr}(x,y)$$

When trimming the camera the two correction constants which depend on the coordinates are determined so that the following is valid:

For:

$$E_e(x,y) = \mathrm{const} \qquad U_{Skorr}(x,y) = \mathrm{const}$$

The breaking down of the correction voltage in two components has proved to be very favourable:

1. Correction voltage (independent of coordinates) with a big setting range for correcting the average value

2. Correction voltage (dependent on coordinates) with a small setting range for correcting the leakage

$$U_{0korr}(x,y) = \overline{U}_{0korr} + \Delta U_{0korr}(x,y)$$

$$U_{1korr}(x,y) = \overline{U}_{1korr} + \Delta U_{1korr}(x,y)$$

The camera calibration is obeyed in 4 seperately selectable steps maximum:

1. Setting of the mean sensor steepness (gain correction voltage, coarse)
2. Setting of the mean absolute sensor brightness (offset correction voltage, coarse)
3. Determination of the correction voltage dependent on coordinates for correcting the steepness (gain correction voltage, fine)
4. Determination of the correction voltage dependent on coordinates for correcting the absolute brightness (offset correction voltage, fine)

1.4 Radiation-physical determinations

To be able to image the irradiance E_e on a numerical range *Z*, an assignment instruction which follows the natural conditions is required.

The following is determined:

1. **The working area of the camera covers an exposure rate of 1:2^{32} (32 octaves)**
2. **This working area shall appear as 10bit number**
3. **The numerical value Z=0 is assigned to the irradiance $E_e=2^{-16}$ *W/m²***

This means the following for the working area:

$$0 \le Z < 1024 \quad \text{for} \quad 2^{-16}\frac{W}{m^2} \le E_e < 2^{16}\frac{W}{m^2}$$

The following relation is produced between the radiation-physical figure and the numerical number Z assigned:

$$E_e = E_{e0} \cdot 2^{\left(\frac{Z}{32}-16\right)} \qquad E_{e0} = 1\frac{W}{m^2}$$

$$Z = 32 \cdot \left[lb\left(\frac{E_e}{E_{e0}}\right) + 16 \right]$$

1.5 Light-technical connection between subject and sensor brightness

When an ideal diffuse reflecting subject is illuminated with the brightness $E_{OB,}$ the subject shines with the brightness L_{OB} of

$$L_{OB} = E_{OB} \cdot \rho \cdot \left(\frac{1}{\pi} \frac{cd}{lx \cdot m^2} \right)$$

r *reflection factor*

The proportional factor $\left(\frac{1}{\pi} \frac{cd}{lx \cdot m^2} \right)$ results from the laws of the ideal diffuse reflection.

The subject is imaged by the lens with an aperture k set on the sensor and with a brightness E_S

$$E_S = L_{OB} \cdot \frac{\tau}{4\left(\frac{k}{k_0}\right)^2} \cdot \left(\pi \frac{lx \cdot m^2}{cd} \right)$$

k *aperture*
k_0 *datum for aperture, =1*

The constant *t* gives the transmittance of the lens. The represenitve figure of ***t = 0,8*** is used in all further calculations.

When putting in the subject brightness $E_{OB,}$ the relation between subject brightness, reflection factor, aperture and sensor brightness is received.

$$E_S = E_{OB} \cdot \frac{\rho \cdot \tau}{4\left(\frac{k}{k_0}\right)^2}$$

Samples for natural brightnesses E_{OB}:

Illuminance	E_{OB} **in [lx]**
Sunlight, summer	100000
Sunlight, winter	10000
Street lighting	3...30
Workroom	40...300
Night with full moon	0,2
Inner rooms	40...150

Reflection factors r for different materials:

Material, object	r
Wood, bright/dark	0,3...0,5/0,1...0,25
Concrete, bright/dark	0,3...0,5/0,15...0,25
Tar cover	0,08...0,15
Brick, bright/dark	0,3...0,4/0,15...0,25
Chromium, polished	0,6...0,7
statistical photografical standard subject	**0,17**

1.6 Relation between radiation-physical and light-technical figures

Light-technical figures take the physiological brightness sensitivity of the human eye into consideration, while radiation-physical figures show the power aspect. To be able to convert on figure into the other one, the relative spectral brightness sensitivity $V(l)$ of the human eye is required. The relation between beam power L_e and luminance L is:

$$L = K_{max} \int_{\lambda=380\,nm}^{780\,nm} L_e(\lambda) \cdot V(\lambda) \cdot d\lambda$$

K_{max} *photo-metrical radiadion equivalent* ***683 lm/W***
$L_e(l)$ *spectral power density of radiation in interval* $(l + dl)$
$V(l)$ *relative spectral brightness sensitivity of the human eye*

l	$V(l)$
400nm	0,0004
500nm	0,323
555nm	1
600nm	0,631
700nm	0,0041

All calculations for calibrating the camera refer to monochromatic light with a wavelength of l *=555nm*. The function $V(l)$ reaches its maximum of *V(555nm)=1* with this wavelength. As exception of the integral mentioned above follows:

$$L = 683 \frac{lm}{W} \cdot L_e$$

The same goes for the intensity of brightness E and the irradiance E_et:

$$E = 683 \frac{lm}{W} \cdot E_e$$

1.7 Calibration equation

The calibration equation gives the connection between the digitally changed numerical value *Z* and the luminance *L* when chosing aperture *k*.

The summing of the equations

$$E_S = L_{OB} \cdot \frac{\tau}{4\left(\frac{k}{k_0}\right)^2} \cdot \left(\pi \frac{lx \cdot m^2}{cd}\right)$$

Conneciton subject luminance – sensor brightness

$$E_S = 683 \frac{lm}{W} \cdot E_{eS}$$

Connection radiation-physical and light-technical figures

$$E_{eS} = E_{e0} \cdot 2^{\left(\frac{Z}{32} - 16\right)}$$

Definition ADC-converting area

results in the calibration equation:

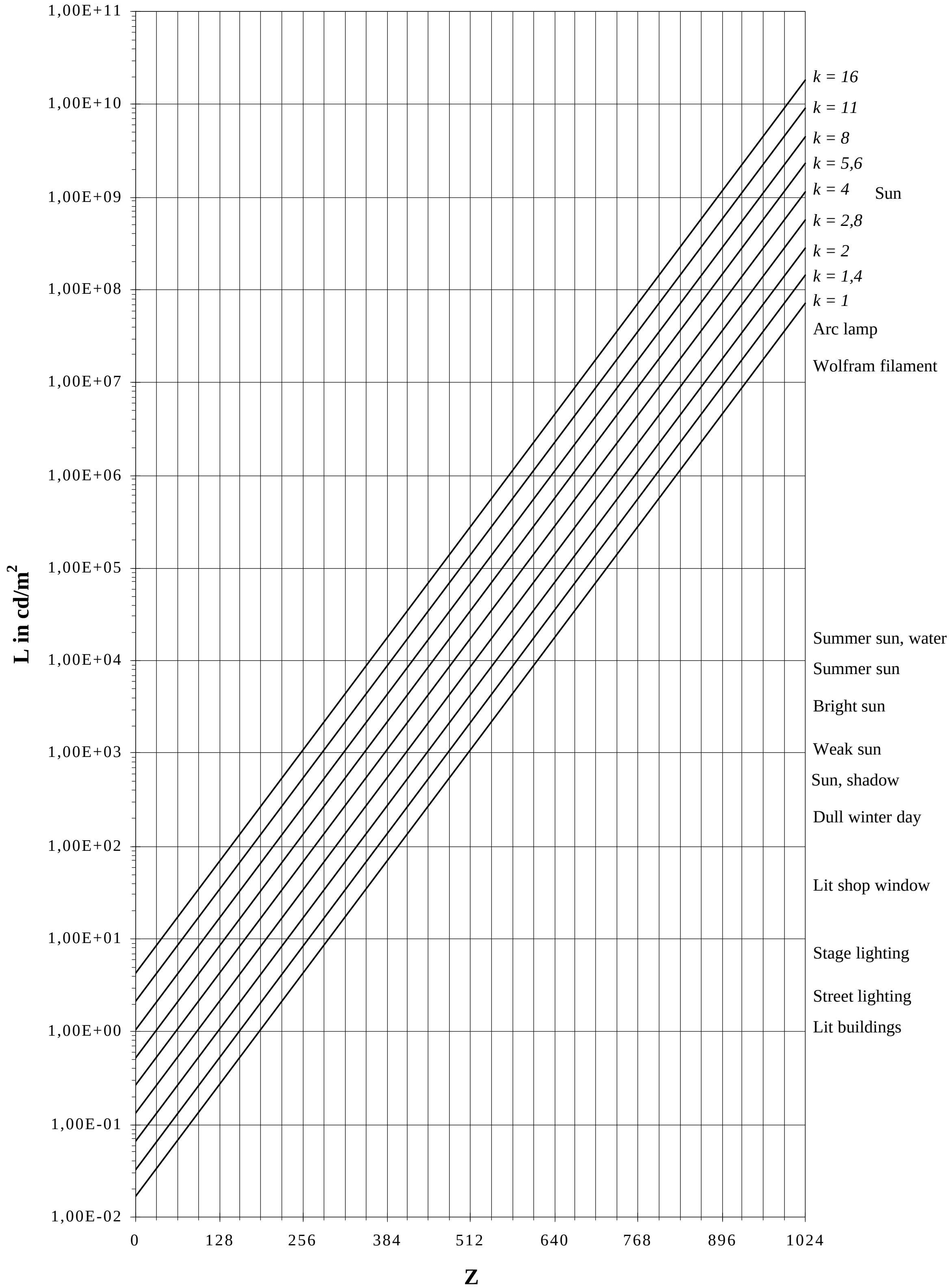
$$\left(\frac{L_{OB}}{L_0}\right) = 683 \cdot \frac{1}{\pi \cdot \tau} \cdot \left(\frac{k}{k_0}\right)^2 \cdot 2^{\left(\frac{Z}{32} - 16\right)}$$

$L_0 = 4 cd/m^2$	*datum for luminance*
$k_0 = 1$	*datum for aperture*
$t = 0{,}8$	*transmittance of the lens*
L_{OB}	*subject luminance in cd/m^2*
Z	*digital ADC numerical value*
k	*aperture*

The calibration equation in logarithmic form:

$$lb\left(\frac{L_{OB}}{L_0}\right) - 2lb\left(\frac{k}{k_0}\right) - lb\left(\frac{683}{\pi \cdot \tau}\right) = \left(\frac{Z}{32} - 16\right)$$

Calibration equation



Presentation of the calibration equation for apertures k=1 to 16

1.8 Presentation of subjects with different brightnesses

When the subject luminance L_{OB} in the logarithmic form of the calibration equation is replaced by the reflection factor r and the subject brightness E_{OB} this results in the following:

$$\mathrm{lb}\left(\frac{E_{OB}}{E_0}\right)+\mathrm{lb}(\rho)-2\mathrm{lb}\left(\frac{k}{k_0}\right)-\mathrm{lb}\left(\frac{683}{\tau}\right)=\left(\frac{Z}{32}-16\right) \qquad E_0=4\mathrm{lx}$$

When illuminating two subjects with reflection factors r_1 and r_2 in succession with two brightnessses E_{OB1} and E_{OB2}, the differences of the values Z are:

Difference of the numerical value DZ between subject with r_1 and r_2 and with brightness E_{OB1} and E_{OB2}:

$$\Delta Z=Z(\rho_1)-Z(\rho_2)=32\cdot\left[\mathrm{lb}(\rho_1)-\mathrm{lb}(\rho_2)\right]$$

Brightness ratios because of different reflection factors r in the subject are given as constant number differences DZ, independent of the subject brightness.

1.9 Assessment of the range of gray scales in the logarithmically altered frame – the photographic normal subject

According to statistical surveys the most often subjects have a contrast difference of 1:32 and an average reflection factor of $r = 0{,}17$. These figures can be consulted when estimating the gray scales Z pictured with a certain brightness.

Definition: photographic standard subject

$$\frac{L_{max}}{L_{min}}=32 \qquad \bar{\rho}=0{,}17$$

The average gray tone results from the above mentioned:

$$\bar{Z}=32\cdot\left[\mathrm{lb}\left(\frac{E_{OB}}{E_0}\right)+\mathrm{lb}(0{,}17)-2\mathrm{lb}\left(\frac{k}{k_0}\right)-\mathrm{lb}\left(\frac{683}{\tau}\right)+16\right]$$

$$\bar{Z}\approx 32\cdot\left[\mathrm{lb}\left(\frac{E_{OB}}{E_0}\right)-2\mathrm{lb}\left(\frac{k}{k_0}\right)\right]+118$$

The difference between the biggest and the smallest gray tone is:

$$\Delta Z=Z_{max}-Z_{min}=32\cdot\mathrm{lb}(32)=160$$

2 The functional architecture of the camera *LOGLUX*Ò

2.1 Survey of functions

The camera *LOGLUX*Ò contains all analog and digital switching components necessary for a frame recording. They can be splitted up into 3 groups:

Analog switching components

1.1 switching components for processing the frame signal (sensor, amplifier)
1.2 switching components for producing an auxiliary voltage

Digital switching components

2.1 switching components for a frame processing (data sorting, address generation)
2.2 switching components for transmitting frame signals (image data interface)
2.3 switching components for camera control (controller and periphery)

AD-, DA-converter

3.1 DA-converter for a digital control of the necessary auxiliary voltages
3.2 AD-converter for converting the image signals

All analog switching components have such a structure that they can be completely digitally balanced by a configuration bus. The analog and digital components necessary for an image recording are controlled by a 16bit-microcontroller. The microcontroller contains all setting features and initializes all assemblies after switching the camera on.

These features of the camera can be gained in different ways:

- automatic identification when trimming the camera
- by transfering instruction over the configuration interface
- by setting the configuration switch on the camera back

The camera has a data interface for transfering the frame datas. This interface is removeable so that the transmission medium and the way of transmission can easily be adjusted.

The functions of the most important components of the HDRC4-camera *LOGLUX*Ò as well as their configura-tion is described in the following:

- **HDRC4 sensor**
  configuration and addressing requirements

- **VSG (Variable Scan Generator)**
  Register description and configuration instructions

- **Data sorting**
  Description of the data sorting depending on the mode

- **DA-converter for producing an auxiliary voltage**
  Register description

2.2 Frame data interface of the camera *LOGLUX*^{Ò}

2.2.1 Description of the interface

Different interface modules can be attached to the camera *LOGLUX*^{Ò}:

- Data transmission via cable, parallel
- LWL-data transmission, serial

The parallel data interface is available in two different variants:

1. LVDS (Low Voltage Differential Signaling) **-** level
2. RS422 **-** level

Both interfaces only differ from each other in their pegel definition but not in the structure of their transmission protocols. These interface standards are symmetric data signals which means that every signal is transmitted once inverted (identification: **-**) and once not-inverted (identification: +). An incorrect data transmission by compensating streams in the signal groundings (application in industrial plants) can thus be avoided.

Bundles of twisted pair two**-**wire circuits with a characteristic impedance of *Z=100W* are used as data cables. For this reason all data receivers require a terminator of *R=100W*.

Comparision RS422, LVDS (with R_L*=100W* , typ.)**:**

Sum of the potential difference between negated and unnegated exit:

$$\Delta U_O = \left| U_{OH} - U_{OL} \right|$$

LVDS	**RS422** (3,3V)
335mV	2,6V

Offset voltage (*Common Mode Voltage)*:

$$U_{CM} = \frac{U_{OH} + U_{OL}}{2}$$

LVDS	**RS422** (3,3V)
1,25V	1,5V

Power demand per signal, at U_b*=3,3V*:

$$P_V = \frac{U_b \cdot \Delta U_O}{R_L}$$

LVDS	**RS422**
11,0mW	85,8mW

Power demand when using all 13 signals (10 data bits, LEN, FEN, CAMCLK):

143mW	**1,1W**

The frame data transmission takes place by using the following signals:

1. CAMCLK (Camera Clock)
2. LEN (Line Enable)
3. FEN (Frame Enable)
4. 10 Bit image datas D0 - D9

The following definitions are valid for the signals LEN, FEN and CAMCLK:

1. The signal LEN („Line Enable“) is defined as HIGH active if the readout of a row is given by the level LEN=H. The same is valid for the signal FEN („Frame Enable“).

2. The signal CAMCLK is defined as HIGH active if the datas being on the exit are valid with an increasing (LH) edge of the signal.

The signals CAMCLK, LEN and FEN can be configured as HIGH or LOW active (LEN, FEN, CAMCLK commands, refer to tabel of commands).

	CAMCLK		**LEN**				**FEN**	
	HIGH active	LOW active	HIGH active		LOW active		HIGH active	LOW active
			lof=0	lof=1	lof=0	lof=1		
Image datas valid	↑ edge	↓ edge	H		L		H	L
Line enable	↑ edge	↓ edge	L		H		H	L
Frame Enable	↑ edge	↓ edge	X	L	X	H	L	H

Tip: meaning *lof*-bit, refer to VSG-Reg. 0, Bit 8

The signal FRAMETRIG can be used for the external frame synchronization (refer to command TRIG, table of commands). The symmetric entry selects a LVDS- (DS90LV032) or RS422- (DS26LV32) receiver and is closed with 100Ω. When the external trigger is switched on (command: TRIG 1) the frame readout process is started with every increasing (LH) edge. After finishing the readout process the camera returns to the waiting state. The status of the FRAMETRIG signal is ignored during the readout process.

2.2.2 Pin-load of the *LOGLUX*® frame data interface

The frame data interface is located on the camera back. (44-pole. D-SUB jack)

Designation	PIN no.	Quty.	Input, Output	Description
D0+	1	1	A	Data bit 0
D1+	2	1	A	Data bit 1
D2+	3	1	A	Data bit 2
D3+	4	1	A	Data bit 3
D4+	5	1	A	Data bit 4
D5+	6	1	A	Data bit 5
D6+	7	1	A	Data bit 6
D7+	8	1	A	Data bit 7
D8+	9	1	A	Data bit 8
D9+	10	1	A	Data bit 9
LEN+	11	1	A	Line Enable
FEN+	12	1	A	Frame Enable
CAMCLK+	13	1	A	Camera Clock
FRAMETRIG+	15	1	E	Frame Trigger
D0-	16	1	A	Data bit 0, negated
D1-	17	1	A	Data bit 1, negated
D2-	18	1	A	Data bit 2, negated
D3-	19	1	A	Data bit 3, negated
D4-	20	1	A	Data bit 4, negated
D5-	21	1	A	Data bit 5, negated
D6-	22	1	A	Data bit 6, negated
D7-	23	1	A	Data bit 7, negated
D8-	24	1	A	Data bit 8, negated
D9-	25	1	A	Data bit 9, negated
LEN-	26	1	A	Line Enable, negated
FEN-	27	1	A	Frame Enable, negated
CAMCLK-	28	1	A	Camera Clock, negated
FRAMETRIG-	30	1	E	Frame Trigger, negated
SDA	31	1	E/A	Datas, I^2C-bus (optional)
SCL	32	1	E	Clock, I^2C-Bus (optional)
GND	33-40	4		Ground
VCC3	41, 42	2	A	3,3V
VCC5	43, 44	2	A	5V

2.2.3 Cable plan, *LOGLUX*Ò frame data interface (feeder cable)

Twisted pair cable, Z≈100Ω, 20 line pairs + screen
2×44-pole D-SUB jack with plug shell

	jack 1, 44-pole D-SUB		jack 2, 44-pole D-SUB	
Designation	Pair no.	Pin no.	Pair no.	Pin no.
D0	1	1	1	1
		16		16
D1	2	2	2	2
		17		17
D2	3	3	3	3
		18		18
D3	4	4	4	4
		19		19
D4	5	5	5	5
		20		20
D5	6	6	6	6
		21		21
D6	7	7	7	7
		22		22
D7	8	8	8	8
		23		23
D8	9	9	9	9
		24		24
D9	10	10	10	10
		25		25
LEN	11	11	11	11
		26		26
FEN	12	12	12	12
		27		27
CAMCLK	13	13	13	13
		28		28
OTR	14	14	14	14
		29		29
FRAMETRIG	15	15	15	15
		30		30
GND1	16	31	16	31
		32		32
GND2	17	33	17	33
		34		34
GND3	18	35	18	35
		36		36
+3,3V	19	41	19	41
		42		42
+5V	20	43	20	43
		44		44
SCHIRM	Housing		-	

2.3 Description of functions: HDRC4-Sensor

The HDRC4-sensor combines on one chip two light-sensitive CMOS-arrays with column and line stucture. Every array has an organisation of 256×256 pixel. The corresponding column and row addresses of both halfs of the sensor are connected. The selection of two pixels (right/left half of the sensor) is possible in different ways. The selection of the access rule is determined by the single bits of the HDRC-register (refer to command *HDRC par* as well). The selection of the half of the sensor is carried out by an additional address bit. This bit selects the seperately digital changed frame information of the left/right half of the sensor via a data multiplexer. Consequently, the address clock frequency can always be half of the pixel clock frequency and two corresponding pixel are read out with one access cycle.

To avoid failures of the analog circuit components caused by the digital components, an addressing in GRAY code is possible. The selection of GRAY or BINARY addressing is determined for both halfs of the senor.

The access rule of the line and column decoder can be selected by an assigned *mirror* bit:

1. *mirror*-bit =0: created column/row address *X* selects column Spalte *X* or row *X*
2. *mirror*-Bit =1: created column/row address *X* selects column or row *(255-X)*

Please note that the mirroring results from a subtraction and not from forming the ones complement because this would lead to a malfunction when having chosen the GRAY addressing.

Determination of the addressing by the HDRC-register:

Bit	7	6	5	4	3	2	1	0
HDRC-Reg.	lrm			gray	Rowlm	collm	rowrm	colrm

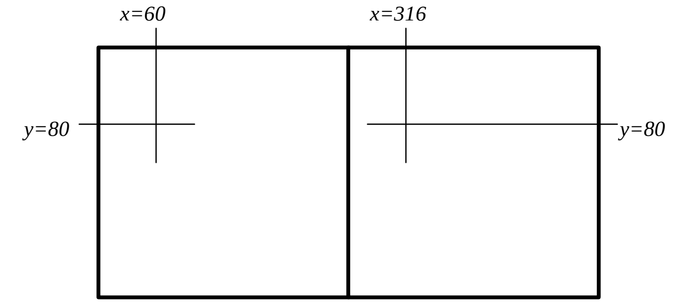
- **colrm**, *column right mirror*: By activating this bit the right-hand half of the sensor can be mirrored around the north-south axis. When addressing column *N*, column *(255-N)* is read out.
- **rowrm**, *row right mirror*: By activating this bit the right-hand half of the sensor can be mirrored around the east-west axis. When addressing line *N*, line *(255-N)* is read out.
- **collm**, *column left mirror*: as colrm, left-hand half of the sensor
- **rowlm**, *row left mirror*: as rowrm, left-hand half of the sensor
- **gray**: Selection of the addressing code, =0 binary code/ =1 gray code.
- **lrm**, *left right mirror*: By activating this bit the addressing of the sensor halfs is interchanged. (When addressing the left-hand half of the sensor, the right-hand half of the sensor is read out and inverted.)

The activating of these bits is independently done by the camera internal control software when using the following commands:

1. Selection readout mode, *MODE*-command
2. Mirroring the image, *MIR*-command
3. Rotating the image, *ROT*-command
4. (initialization of the camera after the switching on)

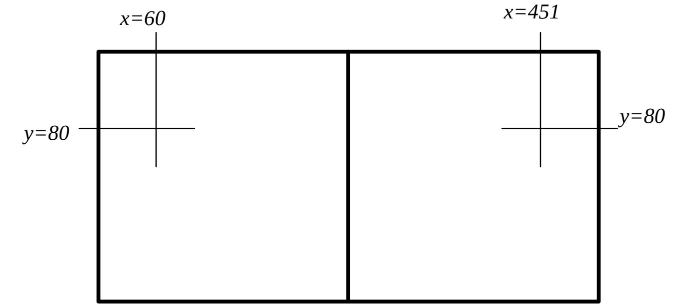
Examples, pixel adressing:

The following examples show the pixel addressing depending on the *mirror*-bits. The sensor provides one pixel formation for the right and left-hand half of the sensor each. All examples are valid for a row address $y = 80$ and a column address $x = 60$.



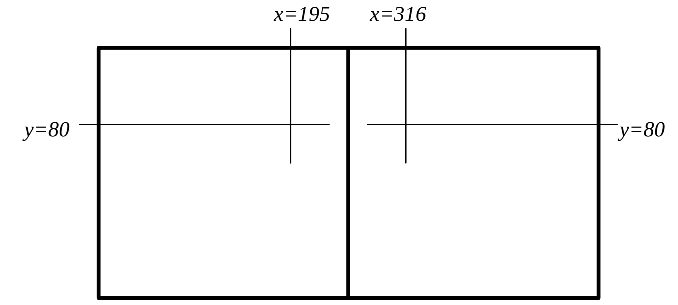
collm =0
rowlm =0

colrm =0
rowrm =0



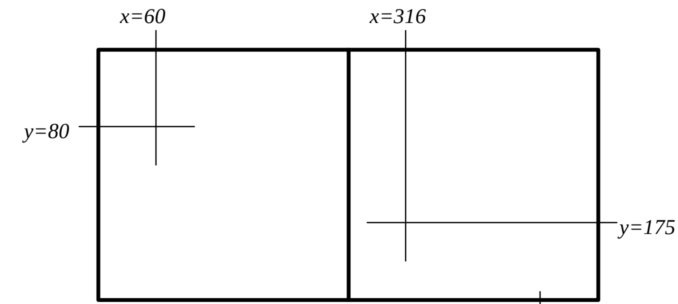
collm =0
rowlm =0

colrm =1, column right mirrored
rowrm =0



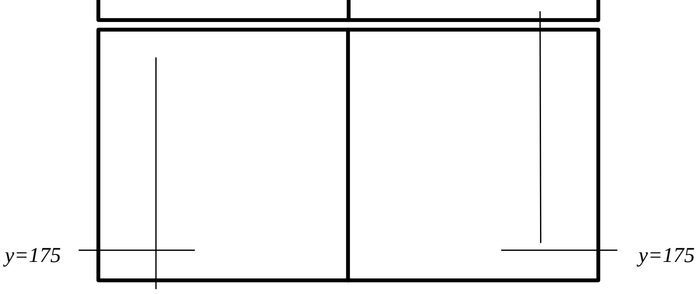
collm =1, column left mirrored
rowlm =0

colrm =0
rowrm =0



collm =0
rowlm =0

colrm =0
rowrm =1, row right mirrored



collm =0
rowlm =1, row left mirrored

colrm =1, column right mirrored
rowrm =1, row right mirrored

2.4 Description of functions: VSG (Variable Scan Generator)

VSG is a programmable address generator. It provides the address, control and synchronous signals required for reading out the sensor:

1. addresses of rows and columns
2. LEN- , FEN-, CAMCLK-signal for frame data transfer
3. Control signals, reading and writing addresses for data sorting

2.4.1 Generation of the column and row addresses

There are two different ways possible for reading out the sensor:

Single channel mode (Mode 0)
The frame information of two pixels is gained in every addressing cycle. However, the information of one pixel is warped afterwards. The order of the pixel coordinates read out of the sensor corresponds to the order of the image datas given out.

Dual channel mode (Mode 2,3)
The image information of two pixels is gained in every addressing cycle. The order of the pixel coordinates read out of the sensor does not correspond to the order of the image datas given out. Therefore an image data sorting is absolutely necessary.

2.4.2 The column and row address counter

VSG contains a 9bit-column and an 8bit-row counter. Start and offset constants of the counter are programmable over register.

The *X/Y_BASE*-register contains the start constants of the respective counter. The resetting is carried out after reaching the count (*X/Y_BASE*) + (*X/Y_OFFSET*).

Example: X_BASE = 100, X_OFFSET = 100
Capacity of the column counter: 100...200 (101 columns!)

X_BASE = 100, X_OFFSET = 100
Capacity of the row counter: 0...255 (256 rows!)

Whole image area:
X_BASE = 0, X_OFFSET = 511
Y_BASE = 0, Y_OFFSET = 255

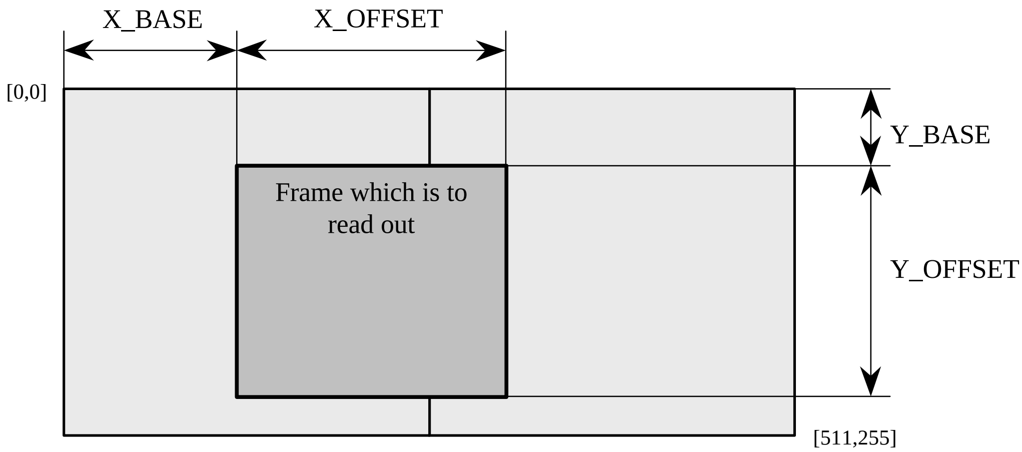
Table, VSG-register 4...7:

Bit	8	7	6	5	4	3	2	1	0
VSG-Reg. 4	x_base (column start constant)								
VSG-Reg. 5		y_base (row start constant)							
VSG-Reg. 6	x_offset (column offset constant)								
VSG-Reg. 7		y_offset (row offset constant)							

2.4.3 Configuration of X/Y_BASE and X/Y_OFFSET-register

2.4.3.1 MODE 0 (single channel mode)

In readout mode 0 the position and size of the frame which is to read out can be selected completely free. The coordinates of the top left-hand frame corner are determined by the X/Y_BASE-register pair, the size of the frame by the X/Y_OFFSET-register pair.



Configuration of the *mirror*-bits for an upright non-reverse image:

colrm = 0
rowrm = 0
collm = 0
rowlm = 0

2.4.3.2 MODE 2 (dual channel mode, converging)

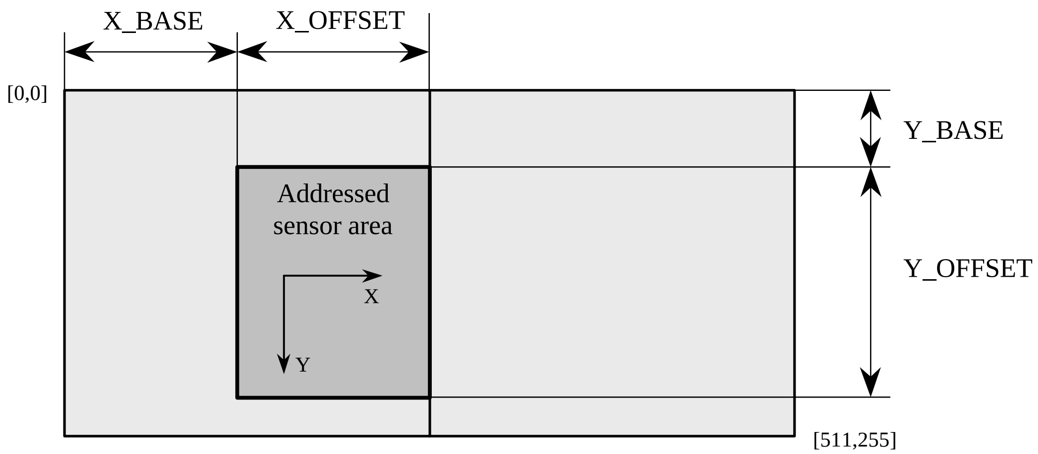
The left-hand sensor half is exclusively addressed in dual channel mode. Corresponding pixel of the right-hand half are read out parallel and afterwards correctly sorted in the image data stream. To get an converging readout, the columns of the right-hand sensor half needs to be mirrored.

Configuration of the *mirror*-bits for an upright and non-reserve image:

colrm = 1 columns right mirrored
rowrm = 0
collm = 0
rowlm = 0

The following sketches show the difference between the addressed and read out sensor area in MODE 2.

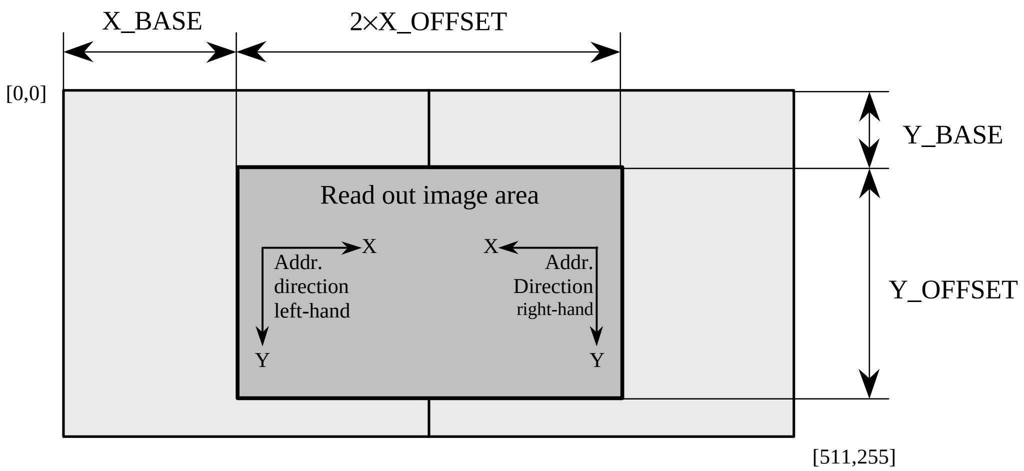
Addressed sensor area:



Secondary requirement in MODE 2: **[X_BASE] + [X_OFFSET] = 255**

Read out sensor area:

nd



2.4.3.3 MODE 3 (dual channel mode, divergent)

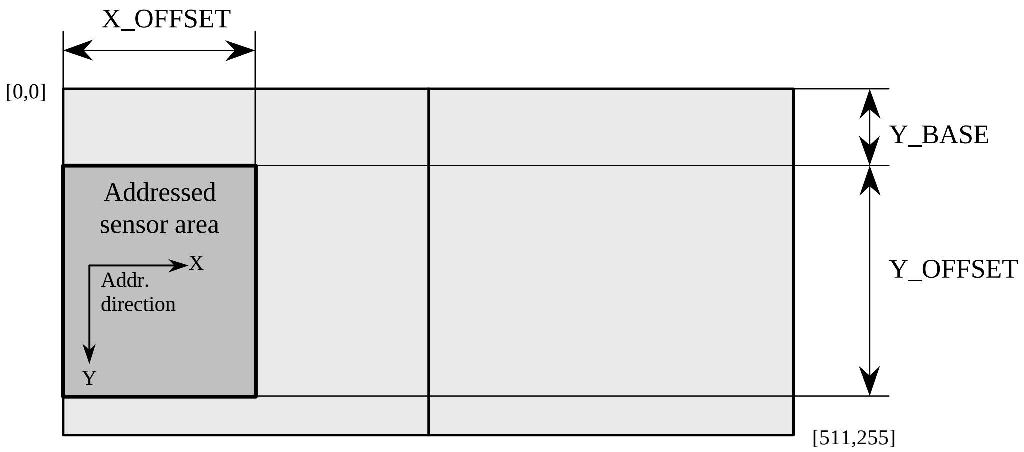
The left sensor half is exclusively addressed in the dual channel mode. Corresponding pixel of the right-hand half are read out parallel and afterwards correctly sorted in the image data stream. To obtain a divergent readout, the columns of the left-hand sensor half have to be mirrored.

Configuration of the *mirror*-bits for an upright and non-reverse image:

colrm	**= 0**	
rowrm	**= 0**	
collm	**= 1**	columns left mirrored
rowlm	**= 0**	

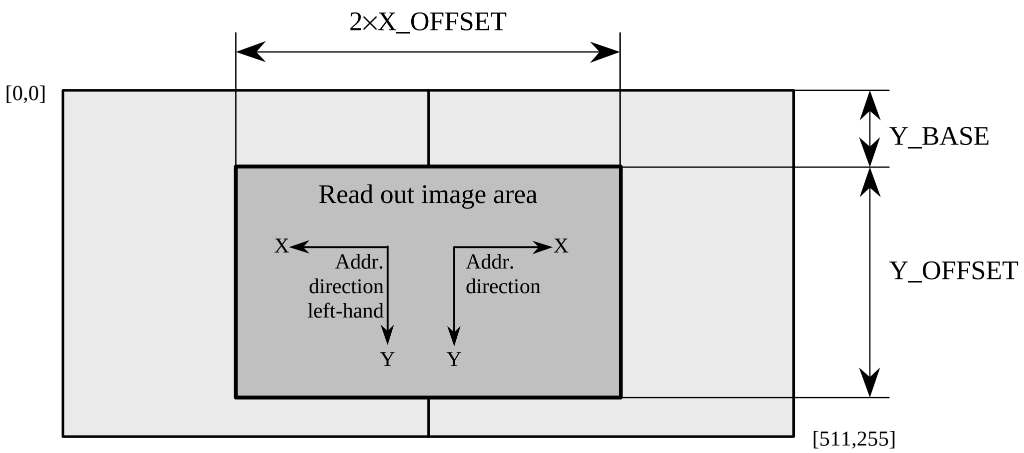
The following sketches show the difference between the addressed and read out sensor area in MODE 3.

Addressed sensor area:



Secondary requirement, MODE 3: **X_BASE = 0**

Read out sensor area:



2.4.4 Configuration of the MODE and PREDIV-registers (VSG-Reg. 0,1)

By configuring the MODE-registers, the readout mode and further essential working parameters of the camera are determined:

- Readout mode
- Pixel clock frequency (f_{camclk})‘
- Single step operating
- External image triggering
- Selection: binary/gray addressing
- OR-concatenation of the synchronous signals
- System-internal settings

Table: MODE and PREDIV-register (VSG-reg. 0,1)

Bit	8	7	6	5	4	3	2	1	0
VSG-Reg. 0	lof	sa	Step	ext_trig	gray	delay	mode		
VSG-Reg. 1								prediv	

Description:

mode (Reg. 0, Bit 0,1,2)

The *mode*-bits determine the readout mode. Only three of all eight possible variants are useful for the programmer (MODE 0,2,3).

MODE 0: *mode* = 000b
MODE 2: *mode* = 010b
MODE 3: *mode* = 011b

delay (Reg. 0, Bit 3)

The delay-bit controls the moment of the row and column address change. It is set by the camera-internal software to **1** and should not be changed.

gray (Reg. 0, bit 4)

=1 row and column addresses are generated in the gray code.
=0 row and column addresses are generated in the binary code.

In order to ensure a correct sensor addressing, the figure of the *gray* bits has to correspond with the *gray* bit of the HDRC-registers. The camera-internal software mainly operates with gray addressing (*gray* = 1).

trig (Reg. 0, bit 5)

=1 single frame mode

A LH single edge on the external triggering entry starts one readout of the sensor with the current X/Y_OFFSET and X/Y_BASE-parameters.

=0 running frame mode

The sensor is continuously read out.

step (Reg. 0, bit 6)

=1 single step operation

The single step operation can only temporary be used by the camera-internal functions.

sa (Reg. 0, bit 7)

camera-internal functions

lof (Reg. 0, bit 8)

=0 The LEN signal is switched „inactive“ during the row synchronization phase.
=1 The LEN signal is switched „inactive“ during the row or frame synchronization phase (*len or fen)*

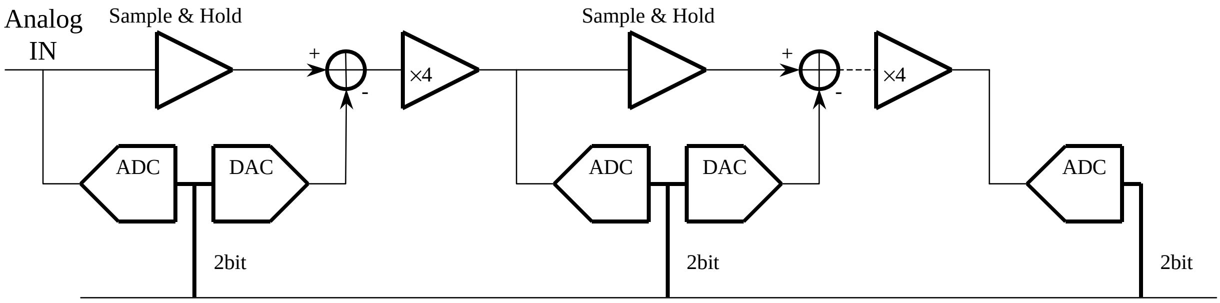
prediv (Reg. 1, bit 0,1)

predivider setting for frame data clock

Prediv	f_{CAMCLK}, MODE 0	f_{CAMCLK}, MODE 2,3
00b	8MHz	16MHz
01b	4MHz	8MHz
10b	2MHz	4MHz
11b	1MHz	2MHz

2.4.5 PIPELINE-DELAY register (VSG reg. 8)

Pipeline-ADC are used for the frame data conversion in LOGLUX cameras. What makes these AD-converter stand out are their low power consumption. The AD conversion is carried out by the operating principle above mentioned in several steps. 2bit information are obtained during every step in this case. A temporal shift of 5 clocks (with 10bit ADC-resolution) between the analog input figure and the converted digital word on the data exit results from this. The sketch shown below gives the architecture of such a cascade transformer in principles.



In order to balance the temporal shift between the sensor addressing and the data sorting, the PIPELINE_ DELAY register has to be loaded with a circuit-specific constant:

PIPELINE_DELAY = [number of clocks required for the conversion]-2

The figure for the PIPELINE_DELAY-register is for the camera *LOGLUX* =5 and must not be changed by the user because this figure is a system constant.

PIPELINE_DELAY = 3

2.4.6 Configuration of the frame data transmission control protocol by the LEN, FEN and MEASURE register

Table: LEN, FEN and MEASURE-register

Bit	8	7	6	5	4	3	2	1	0
VSG-Reg. 2		len							
VSG-Reg. 3		fen							
VSG-Reg. 9		chaninv		feninv	leninv	clkinv	clkoff	pxlclk	adcclk

len (Reg. 2, bit 0...7)

- Length of the inactive phase of the LEN-signal with row synchronization
  Length of the LEN-signal in number *L* CAMCLK-clocks

  MODE 0: **L=[len]**
  MODE 2,3: **L=2×[len]**

fen (Reg. 3, bit 0...7)

- Length of the inactive phase of the FEN-signal with frame synchronization
  Lenght of the FEN-signal in number *Z* rows

  For [fen]=0:
  During the frame synchronization the FEN-signal is in addition to the LEN-signal inactive for the same period of time.

  For [fen]≠0:
  Z=[fen]

adcclk, pxlclk (Reg. 9, bit 0,1)

- Switching off the internal clock signals, set bits always =0

clkoff (Reg. 9, bit 2)

- Switching off the frame clock CAMCLK
  This function is only used camera-internally, set bit always =0

clkinv (Reg.9, bit 3)

- Determination of polarity of the CAMCLK-signal

  clkinv=0:
  frame datas with LH signal edge of the CAMCLK-signal is valid, data change with HL single edge

  clkinv=1:
  frame datas with HL single edge of the CAMCLK-signal is valid, data change with LH single edge

leninv (Reg.9, bit 4)

- Determination of polarity of the LEN-signal

  leninv=0:
  LEN-signal with running row transmission =HIGH, LEN-signal with row synchronization =LOW

  leninv=1:
  LEN-signal with running row transmission =LOW, LEN-signal with row synchronization =HIGH

feninv (Reg.9, bit 5)

- Determination of polaritiy of the FEN-signal

  feninv=0:
  FEN-signal with running frame transimission =HIGH, LEN-signal with frame synchronization =LOW

  feninv=1:
  FEN-signal with running row transmission =LOW, FEN-signal with frame synchronization =HIGH

chaninv (Reg.9, bit 7)

- Inversion of channel selection signal, set bit always =0

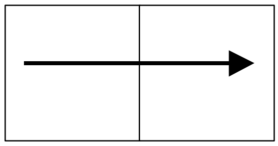
2.5 Frame data sorting

The frame data sorting is carried out by a Dual-Port-RAM 1k×10bit (2 rows×512× 10bit). One port is exclusively used for reading, the other one exclusively for writing. Now VSG provides the sorting addresses in this way that the frame datas read out on the writing port are straight-line written in chronological order and the frame datas of the previous row are straight-line read out in local order on the reading port. Consequently, a time shift of one row can be found between the frame data output and current readout coordinates of the sensor.

2.5.1 Frame data sorting in single channel mode (MODE 0)

When the single channel mode is set, one pixel is read during every addressing cycle. A data sorting is not necessary. However, it is carried out because of circuit-technical reasons. In single channel mode the right half of the sensor is analyzed with a column address ≥256, the left half with a column address <256. The information of the corresponding pixel on the respectively other half of the sensor is rejected. Consequently, a completely free predefinition of the frame which is to read out is possible. However, the maximum pixel clock frequency must not be higher than 8MHz.

The data sorter works from the assumption that the frame datas which need to be sorted are read out sequentially with a rising row and column address.



The frame data sorting is carried out in a simple way:

Sensor access	Dual-Port-RAM, writing port			Dual-Port-RAM, reading port		
	Pixel information Coordinate[X,Y]	Dual-Port-RAM		Pixel information coordinate[X,Y]	Dual-Port-RAM	
		Bank	Writing address		Bank	Reading address
	...	...	...	...	...	...
n	[x ,y]	0	Adr	[x ,y-1]	1	Adr
n+1	[x+1,y]	0	Adr+1	[x+1,y-1]	1	Adr+1
n+2	[x+2,y]	0	Adr+2	[x+2,y-1]	1	Adr+2
n+3	[x+3,y]	0	Adr+3	[x+3,y-1]	1	Adr+3
n+4	[x+4,y]	0	Adr+4	[x+4,y-1]	1	Adr+4
...	...	...	...	...	...	...
m	[x ,y+1]	1	Adr	[x ,y]	0	Adr
m+1	[x+1,y+1]	1	Adr+1	[x+1,y]	0	Adr+1
m+2	[x+2,y+1]	1	Adr+2	[x+2,y]	0	Adr+2
m+3	[x+3,y+1]	1	Adr+3	[x+3,y]	0	Adr+3
m+4	[x+4,y+1]	1	Adr+4	[x+4,y]	0	Adr+4
...	...	...	...	...	...	...

2.5.2 Frame data sorting in dual channel mode (MODE 2,3)

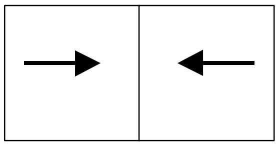
Two corresponding pixel are read in one access cycle in the dual channel mode. MODE 2 and MODE 3 differ from each other only in their sorting algorithm by the interpretation of the coordinates of the sequentially read frame datas.

2.5.2.1 MODE 2:

The frame datas are sorted in this way that the frame datas are sorted locally correct when being convergingly read out. Converged means in this case that corresponding pixel pairs draw nearer to each other during the readout process.

Example, converging readout:
Pixel coordinates in one cycle, full frame:

n: {[x,y] , [511- x,y]}
n+1: {[x+1,y] , [511-(x+1),y]}
n+2: {[x+2,y] , [511-(x+2),y]}
n+3: {[x+3,y] , [511-(x+3),y]}
n+4: {[x+4,y] , [511-(x+4),y]} etc.



mirror-bits: colrm = 1
rowrm = 0
collm = 0
rowlm = 0

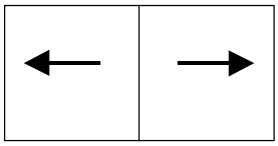
Sensor access	Dual-Port-RAM, writing port			Dual-Port-RAM, reading port		
	Pixel information coordinate[X,Y]	Dual-Port-RAM		Pixel information Coordinate[X,Y]	Dual-Port-RAM	
		Bank	Writing address		Bank	Reading address
	...	...	...	...	...	...
n	[0,y]	0	0x000	[0,y-1]	1	0x000
	[511,y]	0	0x001	[1,y-1]	1	0x002
n+1	[1,y]	0	0x002	[2,y-1]	1	0x004
	[510,y]	0	0x003	[3,y-1]	1	0x006
n+2	[2,y]	0	0x004	[4,y-1]	1	0x008
	[509,y]	0	0x005	[5,y-1]	1	0x00A
n+3	[3,y]	0	0x006	[6,y-1]	1	0x00C
	[508,y]	0	0x007	[7,y-1]	1	0x00E
n+4	[4,y]	0	0x008	[8,y-1]	1	0x010
	[507,y]	0	0x009	[9,y-1]	1	0x012
	...	...	...	...	...	...
n+126	[126,y]	0	0x0FC	[252,y-1]	1	0x1F8
	[385,y]	0	0x0FD	[253,y-1]	1	0x1FA
n+127	[127,y]	0	0x0FE	[254,y-1]	1	0x1FC
	[384,y]	0	0x0FF	[255,y-1]	1	0x1FE
n+128	[128,y]	0	0x100	[256,y-1]	1	0x1FF
	[383,y]	0	0x101	[257,y-1]	1	0x1FD
n+129	[129,y]	0	0x102	[258,y-1]	1	0x1FB
	[382,y]	0	0x103	[259,y-1]	1	0x1F9
n+130	[130,y]	0	0x104	[260,y-1]	1	0x1F7
	[381,y]	0	0x105	[261,y-1]	1	0x1F5
...	...	...	...	...	...	...
n+255	[255,y]	0	0x1FE	[510,y-1]	1	0x003
	[511,y]	0	0x1FF	[511,y-1]	1	0x001
n+256	[255,y+1]	1	0x000	[0,y]	0	0x000
	[256,y+1]	1	0x001	[1,y]	0	0x002
...	...	1	...	...	0	...

2.5.2.2 MODE 3:

The frame datas are sorted in this way that frame datas are sorted locally correct when the sensor is read out divergently. Divergent means here that corresponding pixel pairs approach to the respective edge of the frame when being read out.

Example, divergent readout:
Pixel coordinates in one cycle, full frame:

n: {[255-x,y] , [256+x,y]}
n+1: {[255-(x+1),y] , [256+(x+1),y]}
n+2: {[255-(x+2),y] , [256+(x+2),y]}
n+3: {[255-(x+3),y] , [256+(x+3),y]}
n+4: {[255-(x+4),y] , [256+(x+4),y]} u.s.w



mirror-bits: colrm = 0
rowrm = 0
collm = 1
rowlm = 0

Sensor access	Dual-Port-RAM, writing port			Dual-Port-RAM, reading port		
	Pixel information coordinate[X,Y]	Dual-Port-RAM		Pixel information coordinate[X,Y]	Dual-Port-RAM	
		Bank	Writing address		Bank	Reading address
	...	...	...	...	...	...
n	[255,y]	0	0x000	[0,y-1]	1	0x1FE
	[256,y]	0	0x001	[1,y-1]	1	0x1FC
n+1	[254,y]	0	0x002	[2,y-1]	1	0x1FA
	[257,y]	0	0x003	[3,y-1]	1	0x1F8
n+2	[253,y]	0	0x004	[4,y-1]	1	0x1F6
	[258,y]	0	0x005	[5,y-1]	1	0x1F4
n+3	[252,y]	0	0x006	[6,y-1]	1	0x1F2
	[259,y]	0	0x007	[7,y-1]	1	0x1F0
n+4	[251,y]	0	0x008	[8,y-1]	1	0x1EE
	[260,y]	0	0x009	[9,y-1]	1	0x1EC
	...	...	...	...	...	...
n+126	[129,y]	0	0x0FC	[252,y-1]	1	0x006
	[382,y]	0	0x0FD	[253,y-1]	1	0x004
n+127	[128,y]	0	0x0FE	[254,y-1]	1	0x002
	[383,y]	0	0x0FF	[255,y-1]	1	0x000
n+128	[127,y]	0	0x100	[256,y-1]	1	0x001
	[384,y]	0	0x101	[257,y-1]	1	0x003
n+129	[126,y]	0	0x102	[258,y-1]	1	0x005
	[385,y]	0	0x103	[259,y-1]	1	0x007
n+130	[125,y]	0	0x104	[260,y-1]	1	0x009
	[386,y]	0	0x105	[261,y-1]	1	0x00B
...	...	...	...	...	...	...
n+255	[0,y]	0	0x1FE	[510,y-1]	1	0x1FD
	[511,y]	0	0x1FF	[511,y-1]	1	0x1FF
n+256	[255,y+1]	1	0x000	[0,y]	0	0x1FE
	[256,y+1]	1	0x001	[1,y]	0	0x1FC
...	...	1	...	...	0	...

3 Configuration of the camera *LOGLUX*Ò

3.1 Introduction

The device-internal camera control makes it possible to set the following parameters over the configuration interface:

- frequency and polarity of the camera clock CAMCLK
- pulse width and polarity of the row synchronuous signal LEN (Line Enable)
- pulse width and polarity of the row synchronuous signal FEN (Frame Enable)
- size and position of sensor half which is to read out
- readout mode
- infensification and offset of the video amplifier
- selection of the correction table

In order to make the camera configuration easier for the user, the internal software offers two opportunities:

1. ASCII (plain text) control
2. HEX control

The selection of the way of controlling and of the interface-specific parameter is effected by the configuration switch on the camera back.

*LOGLUX*Ò cameras have a RS232C (V.24) configuration interface as standard. All parameters declared are not erased when switching the camera off and are automatically set when switching the camera on or when pressing the RESET button. It is, therefore, possible to use the camera independently of a configuration computer or to do the configuration only when installing the camera.

Requirements for the camera configuration:

1. Configuration computer with serial interface (e.g. COM1, PC)
2. VT100 terminal or special configuration software
3. 1 zero modem cable

3.2 RS232C Configuration interface

The camera-internal RS232C interface is equipped with an IC MAX3232 and attends the signals TxD, RxD, CTS and RTS. The signals DTR, DSR and DCD are internally connected. For controlling the data transmission the hardware (RTS/CTS) control protocol is applied. The transmission rate, the number of stop bits and data bits as well as the parity bit can be selected with the configuration switch (see below). At the same time it is possible to define the interface parameters completely free or from a default selection.

The 9-pole D-SUBm plug located on the camera back has the following pin assignments:

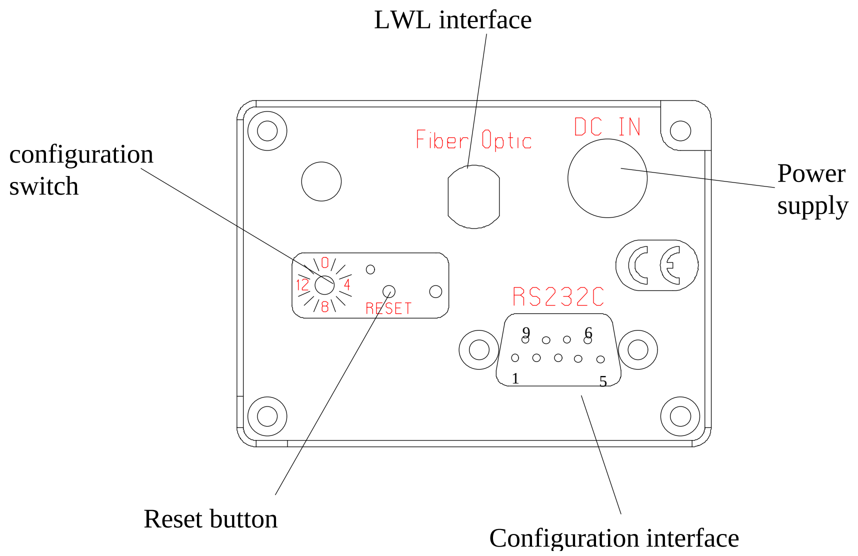
RS232 connection, 9-pole D-SUB plug

Designation	PIN #,	Quantity	Input/Output	Description
/DCD	1	1	E	DATA CARRIER DETECTED with /DTR and /DSR connected
RxD	2	1	E	Receive datas
TxD	3	1	A	Transmission datas
/DTR	4	1	A	DATA TERMINAL READY with /DCD and /DSR connected
GND	5	1		GROUND
/DSR	6	1	E	DATA SET READY with /DCD and /DTR connected
/RTS	7	1	A	REQUEST TO SEND
/CTS	8	1	E	CLEAR TO SEND
RI	9	1	E	RING INDICATOR Not connected

A cable (zero modem cable) with the connections mentioned below is necessary to connect the camera with the configuration computer.

List of connections for the zero modem cable

Jack 1 9-pole D-SUBw	RxD PIN 2	TxD PIN 3	RTS PIN 7	CTS PIN 8	DTR-DCD PIN 4-1	DSR PIN 6	GND PIN 5
Jack 2 9-pole D-SUBw	TxD PIN 3	RxD PIN 2	CTS PIN 8	RTS PIN 7	DSR PIN 6	DTR-DCD PIN 4-1	GND PIN 5



back side of the camera

3.3 RS422 Configuration interface (optionally)

The configuration interface can optionally be equipped with a RS422 driver (MAX3488). The interface operates without a hardware control protocol in this case (only RxD+, RxD-, TxD+, TxD-). The D-Sub plug was wired in this way that a zero modem cable of a RS232C interface can further be used for communication.

To make a data transmission over longer distances possible, the symmetric data input (RxD+, RxD-) is closed with a terminator of *R = 100W*.

RS422 connection, 9-pole D-SUB jack

Designation	PIN #	Quantity	Input/Output	Description
	1	1		Connected with PIN 4,6
RxD+	2	1	I	Receive datas
TxD+	3	1	O	Transmission datas
	4	1		Connected with PIN 1,6
GND	5	1		GROUND
	6	1		Connected with PIN 4,1
RxD-	7	1	I	Receive datas, inverted
TxD-	8	1	O	Transmission datas, inverted
	9	1		Free

3.4 Signal-LED

The signal-LED on the camera back gives information about the present operating condition of the camera. The following signals are assigned to different conditions:

Signal	**Operating condition**
Green continuous light	camera operates, commands cannot be received (configuration switch position 15)
Green continuous light, short yellow blinking (0,5Hz)	camera operates, commands can be received (configuration switch position 0...14)
Green flashing light	camera carries out the command or is in the base initialization after the switching on
Yellow flashing light, slowly	Camera trimming
Yellow flashing light, quickly	Flash-ROM is programmed
Yellow/red flashing light	camera requests during the trimming routine a high luminance
Yellow/green flashing light	camera requests during the trimming routine a low luminance
Green/red flashing light	Base initialization after the *WATCHDOG* reset

3.5 The Configuration switch

The configuration switch is used for selecting the

1. kind of control (plain text, HEX mode)
2. interface parameter.

Together with the RESET button the configuration switch is located on the camera back under a base plate.

The position of the configuration switch is enquired when powering up the operating voltage or after RESET i.e. an adjusting under running does not have immediate influence. If the camera is equipped with the standard configuration switch (RS232C), the following interface and configuration parameter arise for the different switch positions:

Position	Configuration mode	Transmission rate	Datas		
0	Plain text m.	1200 Bd	8 Bit	No parity	1 Stop bit
1		1200 Bd	7 Bit	No parity	
2		9600 Bd	8 Bit	No parity	
3		9600 Bd	7 Bit	No parity	
4	HEX mode	1200 Bd	8 Bit	No parity	
5		1200 Bd	8 Bit	Even	
6		1200 Bd	8 Bit	Uneven	
7		9600 Bd	8 Bit	No parity	
8		9600 Bd	8 Bit	Even	
9		9600 Bd	8 Bit	Uneven	
10		19200 Bd	8 Bit	No parity	
11		19200 Bd	8 Bit	Even	
12		19200 Bd	8 Bit	Uneven	
13	Plain text m.	User-defined			
14	HEX mode	User-defined			
15	No external configuration (conf. Interface is not initialized)				

Example: *plain text control 9600Bd, 8bit, no parity:* switch position

3.6 Camera control in ASCII (plain text) mode

If the configuration switch is in setting 0...3 or 13, the camera can be configured in plain text. The camera has to be connected to a VT100 compatible terminal for this (TERMINAL WIN3.1x, HYPER TERMINAL WIN95, TERMINAL VT100 Norton Commander) and the terminal software needs to be configured as follows:

- Transmission rate, 7/8 bit, parity, stop bit → see table, configuration switch
- Hardware control protocol (RTS/CTS)
- No local echo
- Function keys for Windows OFF!

All commands are input in plain text. The data sender (keyboard) operates independently of the receiver because the camera transmits a character echo itself. Small letters are automatically shifted into capital letters. All printable ASCII characters (0x20...0x7F) are analyzed as well as the following control characters and ESCAPE series:

^M	0x0D (carriage return, CR)	RETURN button
ESC O P	0x1B / 0x4F / 0x50	F1 button
ESC O Q	0x1B / 0x4F / 0x51	F2 button
ESC O P	0x1B / 0x4F / 0x52	F3 button
ESC O Q	0x1B / 0x4F / 0x53	F4 button
ESC [A	0x1B / 0x5B / 0x41	↑ button
ESC [B	0x1B / 0x5B / 0x42	↓ button
ESC [C	0x1B / 0x5B / 0x43	→ button
ESC [D	0x1B / 0x5B / 0x44	←-Taste

The data receiver (display) must be able to process the control characters given below in addition to all printable ASCII characters (0x20...0x7F):

^M	0x0D	(carriage return, CR)	Cursor to the beginning of the line
^H	0x08	(back space, BS)	Cursor one character to the left
^J	0x0A	(line feed, LF)	Cursor one character downwards
^G	0x07	(bell, BEL)	Acoustic signal

Layout of function keys:

F1:	BACKUP,	Saving the present configuration
F2:	UNDO,	Setting of the configuration saved
F3:	Display of the configuration	
F4:	Single frame statistics	

Commands and parameters are seperated by blanks, parameters from each other by a comma:

COMMAND   [parameter1 [, parameter2]]

Example: CAMCLK 16,0

When using the terminal function „SEND TEXT FILE“, it is possible to send command groups (e.g. setting for certain types of framegrabbers). The file which is to be sent can be created with any editor (e.g. EDIT, WRITE).

Example: Setting of the camera clock signal for ELTEC framegrabber PCEye with CAMA 160:

```
// configuration ELTEC framegrabber
len 16,0
fen 0,0
// pixel clock 4MHz
camclk 4,0
```

3.7 Camera control in HEX mode

Is the configuration switch in setting 4...12 or 14, the camera can be configured in plain text mode. This mode permits the control with a minimum of characters transmitted. Every command has a capacity of 4 byte maximum. That is why, the configuration interface has to be programmed to a transmission of 8-bit data capacity. Furthermore, it is possible to transmit command sequences as datagrams which are sequentially processed.

A datagram (command sequence) has always the following structure:

Byte 0	Byte 1	Byte 2	Byte 3		Byte n-1	Byte n
(Sequence length n)	date 1	date 2	date 3		date n-1	data n

Byte 0 includes the length of the command sequence. (An empty sequence has the sequence length 0)

Structure of an command:

1 byte command

Command code

2 byte command

Command code	Date 1

3 byte command

Command code	Date 1	Date 2

4 byte command

Command code	Date 1	Date 2	Date 3

The length of the instruction results from the command code. Is an command or an command sequence processed, a return data sequence consisting of 1 byte minimum (only error code) is produced. Is an command sequence processed, the following command of this sequence is only then obeyed if this command produces the error code 0x00 (ok). In case of an error the process is interrupted and the error code is sent to the control computer via the configuration interface.

Table: Error codes

Error code		Description
Decimal	Hex.	
0	0x00	Command is properly obeyed
1...127	0x01...0x7F	Marking byte, m data bytes follow (see table below)
128	0x80	Command sequence too long
249	0xF9	Set frame format or frame position cannot be adjusted in the selected readout mode
250	0xFA	Set pixel clock frequency cannot be adjusted in the selected readout mode
251	0xFB	Parameter is illegal in the selected readout mode
252	0xFC	Priviliged command, see $ command
253	0xFD	Illegal parameter
254	0xFE	Number of parameters too small
255	0xFF	Illegal command code

If the command includes the sending of datas to the control computer (e.g. enquiring the software version), these datas are sent before the respective error code (marking by the error code 1..127). The sent marking byte corresponds to the error code of the command which effects the sending of further data bytes. The number of data bytes following the marking byte comes from the marking byte in accordance with the following assignment:

Assignment marking byte (=command code) **-** number ***m*** of the following data bytes

Marking byte	number ***m*** of the following data bytes
0x01 (VERSION)	4
0x0F (EEPROM)	128
0x17 (ADC)	2
0x1B (STAT)	30

A definite analysis of a return data sequence is possible according to the following diagramm:

```
Send command sequence

cancel=no

DO
        Input data byte date

        IF (date<128 or date>0)
                Determine from date number*)
                Input number of data bytes                (read return datas)

        ELSE    IF (date=0)
                        result=ok
                ELSE
                        result=error

                cancel=yes

WHILE(abbruch=nein)

Analyze result
```

*) This function assigns the length of the return sequence to a command. (see command description)

Example: Command sequence sent with **proper** process

Plain text: *version*
mode 3

- Command sequence, sent byte series:

Byte 0	Byte 1	Byte 2	Byte 3
0x03	0x01	0x09	0x03
Command length	Command code VERSION	Command code MODE	Parameter „3“

- Return data sequence, received byte series

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x01	0x00	98	03	24	0x00
Marking byte, Byte signalizes: 4 data bytes follow	Data byte0 IDENTIFICA-TION „*LOGLUX*“	Data byte1 YEAR	Data byte2 MONTH	Data byte3 DAY	Error code for whole sequence: „ok“

Sent command sequence with **incorrect** parameters:

Plain text: *mode 72* „72“ uncorrect parameter
version

- Command sequence, sent byte series:

Byte 0	Byte 1	Byte 2	Byte 3
0x03	0x09	72	0x01
Command length	Command code MODE	Parameter „3“	Command code VERSION

- Return data sequence, received byte series

Byte 0
0xFD
Error code „illegal parameter“

The command VERSION was not obeyed because the command MODE was cancelled by an incorrect parameter.

4 Description of commands

4.1 Table of commands

According to their functions the commands given below can be divided into the following groups:

- Commands for determining the frame data transmission

  CAMCLK
  LEN
  FEN
  TRIG

- Commands for determining the frame size and position

  RESET
  FRAME_SIZE
  FRAME_POS
  ROT
  MIR

- Matching and test commands

  HIGH
  LOW
  CAL
  WR

- Enquiring the camera status

  VERSION
  EEPROM
  ADC

- Register commands

  VSG
  MUX
  HDRC
  DAC

4.2 Structure of the command description:

COMMAND	Name of the command for plain text mode
• Abstract	Abstract of the commands
0x00	Coding of the command in HEX mode
Return: Error code	Return sequence
Length of the command: 1 byte	Command length

RESET

- Reset of all camera settings

0x00

Return:

Error code

Length of command: 1byte

The RESET command allocates all variable configuration parameter with a defined value and re-initializes the correction RAM.

When calling the RESET command the following MACRO is started:

```
DAC     0, 150
DAC     1, 140
DAC     2, 128
DAC     3, 128
MODE    0
TAB     4
```

Configuration:

Bit	8	7	6	5	4	3	2	1	0
HDRC reg.		lrm **1**	resdir **1**	nores **1**	gray **1**	rowlm **0**	Collm **0**	rowrm **0**	colrm **0**
VSG reg. 0	lof **0**	sa **0**	step **0**	ext_trig **0**	gray **1**	delay **1**	mode **0**		
VSG reg. 1								prediv **1**	
VSG reg. 2		len **16**							
VSG reg. 3		fen **0**							
VSG reg. 4	x_base **0**								
VSG reg. 5		y_base **0**							
VSG reg. 6	x_offset **511**								
VSG reg. 7		y_offset **255**							
VSG reg. 8							pipedelay **3**		
VSG reg. 9		chaninv **0**		feninv **1**	leninv **1**	clkinv **0**	clkoff **0**	pxlclk **0**	adcclk **0**

DAC reg.0	gain **150**
DAC reg.1	offset **140**
DAC reg.2	adcref **128**
DAC reg.3	xxx **128**

Correctur table	tab (8bit) **4**

The contents of the DAC-Backup register and the matching references remain unchanged.

VERSION

• Enquiring the software version (date of compiling)

0x01

Length of the command: 1byte

Return: 6 byte

0x01	0x00	year	month	day	error code

Error code: [0] ok

The contents of all configuration registers remains unchanged.

$

• Release of the extended command set

0x02

Length of the command: 1byte

Return: 1 byte

Error code

The extended command set is released by the **$** command. This makes it possible to have a direct access to the configuration register. The use of the extended command set requires the exact knowledge of the camera control because incorrect parameters may cause a malfunction of the camera.

The contents of all configuration registers remains unchanged.

If there is any access to a command of the extended command set (privileged command) without prior release, the command sequence is cancelled and acknowledged with error code [252].

DAC channel, parameter

(privileged command, see $ command)

• Setting the DAC channel

0x03	channel	parameter

Length of the command: 3 bytes

Return: 1 byte

Error code

channel: number of the channel (0...3)

channel 0: GAIN register
channel 1: OFFSET register
channel 2: ADCREF register (HIGH reference video ADC)

parameter: 0...255

channel 0:	GAIN register	standard parameter:	150
channel 1:	OFFSET register	standard parameter:	140
channel 2:	ADCREF register	standard parameter:	128

error code:

[0] ok
[252] access to privileged command (see **$** command)

GAIN register: The intensification of the video amplifier results from the following equation:

$$V(dB) = 20dB \cdot \left(value \cdot 6{,}78 \cdot 10^{-3} + 0{,}381\right) - 19dB$$

$$\Delta V(db) = \Delta value \cdot 0{,}136dB$$

Examples:

Value	0	100	**150**	200	255
V(dB)	-11,4dB	2,2dB	**9,0dB**	15,8dB	23,2dB

ADCREF register: The difference between HIGH and LOW reference (range of converter) of the video-ADC results from the following equation:

$$\Delta U = 312mV + value \cdot 5{,}56mV)$$

Examples:

Value	100	**128**	150	200	255
U	868mV	**1024mV**	1146mV	1424mV	1730mV

MUX channel, switch

(privileged command, see $ command)

• Setting of a multiplex channel

0x04	channel	parameter

Length of the command: 3 byte

Return: 1 byte

Error code

channel: number of the channel (0, 1)

schalter: switch position (0...3)

error code:	[0]	ok
	[252]	access to privileged commands (see **$** command)

The MUX command activates the camera-internal analog multiplex channel. This command can be used for test purposes only and has no influence on the camera configuration.

HDRC parameter (privileged command, see $ command)

• Loading the HDRC register

0x05	parameter

Length of the command: 2 byte

Return: 1 byte

Error code

parameter: parameter of the HDRC register (0...255)

error code:	[0]	ok
	[252]	access to privileged command (see **$** command)

The HDRC command sets the HDRC register. If bit 0...4 differs between old and new register parameter, the correction RAM is additionally re-configured. The selection of the table depends on the TAB register. (See TAB command)

Configuration by HDRC command:

Bit	7	6	5	4	3	2	1	0
HDRC reg.	lrm ⇔	Resdir ⇔	nores ⇔	gray ⇔	rowlm ⇔	collm ⇔	rowrm ⇔	colrm ⇔

Correction table	tab (8bit) ×

Structure of the HDRC register:

- **colrm**, *column right mirror*: By setting this bit the right-hand half of the sensor can be mirrored around the north-south axis. When addressing column (n), column (255-n) is read out.
- **rowrm**, *row right mirror*: By setting this bit the right-hand half of the sensor can be mirrored around the east-west axis. When addressing row (n), row (255-n) is read out.
- **collm**, *column left mirror*: as colrm, left-hand half of the sensor
- **rowlm**, *row left mirror*: as rowrm, left-hand half of the sensor
- **gray**: Selection of the addressing code, =0 binary code/ =1 gray code. This bit is tightly connected with the **gray** bit VSG reg. 0, bit 4
- **nores**: Suppresses the reset procedure which is processed row by row. The parameter of this bit should not be changed by the user.
- **resdir**: Determines the reset procedure. The parameter of this bit should not be changed by the user.
- **lrm**, *left right mirror*: By setting this bit, the addressing of the sensor halfs is interchanged. (When addressing the left-hand half of the sensor, the right-hand half of the sensor is read out and inverted.)

(×: value is not changed ⇔: value is changed according to parameter **0**: value = 0 etc.)

VSG reg, parameter

(privileged command, see $ command)

• Loading of a VSG register

0x06	reg	high (parameter)	low (parameter)

Length of the command: 4 byte

Return: 1 byte

Error code

Error code: [0] ok
[252] access to privileged commands (see **$** command)

high(parameter): High byte register parameter (0, 1)
low(parameter): Low byte register parameter (0, 255)

reg: register no. (0...9)

Configuration by VSG command:

Bit	8	7	6	5	4	3	2	1	0
VSG reg. reg	⇔	⇔	⇔	⇔	⇔	⇔	⇔	⇔	⇔

VSG configuration register:

VSG register	reg
MODE register	0
PREDIV register	1
LEN register	2
FEN register	3
X_BASE register	4
Y_BASE register	5
X_OFFSET register	6
Y_OFFSET register	7
PIPEDELAY reg.	8
MEASURE reg.	9

Refer to chapter „2.4 Descriptions of function VSG“ for an exact description of the VSG register.

FRAME_SIZE X, Y

• Determination of the frame area

0x07	high (X)	low (X)	Y

Length of the command: 4 byte

Return: 1 byte

Error code

Erroe code: [0] ok

high(X): (High byte X coordinate)-1 (0, 1)
low(X): (Low byte X coordinate)-1 (0...255)

Y: (Y coordinate)-1 (0...255)

Example: FRAME_SIZE 199,99 frame area (200×100) pixel

The command FRAME_SIZE determines the frame area which needs to be read out and which depends on the readout mode set (see command MODE). The following restrictions are valid for this with reference to the area set:

- MODE 2,3: north-south axial-symmetric frame areas are possible only, even X coordinates only
- MODE 0: the selection of the frame area is completely free

Configuration by the FRAME_SIZE command:

Bit	8	7	6	5	4	3	2	1	0
VSG reg. 0	lof ×	sa ×	step ×	ext_trig ×	gray ×	delay ×	mode ×		
VSG reg. 1								prediv ×	
VSG reg. 2		len ×							
VSG reg. 3		fen ×							
VSG reg. 4	x_base ⇔/× (see table)								
VSG reg. 5		y_base ×							
VSG reg. 6	x_offset ⇔ (see table)								
VSG reg. 7		y_offset ⇔ (see table)							
VSG reg. 8							pipedelay ×		
VSG reg. 9		chaninv ×		feninv ×	Leninv ×	Clkinv ×	clkoff ×	pxlclk ×	adcclk ×

(×: value is not changed ⇔: value is changed according to parameter **0**: value = 0 etc.)

VSG-REGISTER	MODE 0	MODE2	MODE 3
4 (X_BASE)	×(not changed)	255-X/2 *	0
6 (X_OFFSET)	X	X/2 *	X/2 *
7 (Y_OFFSET)	Y	Y	Y

* The LSB of the X coordinate is lost when dividing by 2. That is why, a frame area of (300x200) pixel is set each time when applying the following commands:

FRAME_SIZE 299, 199
FRAME_SIZE 298, 198

FRAME_POS X, Y

- Determination of the frame position

0x08	High (X)	low (X)	Y

Length of the command: 4 byte

Return: 1 byte

Error code

Error code: [0] ok

high(X): (High byte X coordinate) MODE 0: (0, 1) MODE 1,2,3: (0)
low(X): (Low byte X coordinate) (0...255)

Y: (Y coordinate) (0...255)

The command FRAME_SIZE determines the position of the upper left-hand corner of the image. This depends on the readout mode set (see command MODE). The following commands are valid with reference to the varying coordinates:

- MODE 2,3: Y coordinate is variable only (0...255), X coordinate is ignored
- MODE 0: the selection of the frame position is completely free

Configuration by the FRAME_POS command:

<table>
<tr><th>Bit</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>VSG reg. 0</td><td>Lof
×</td><td>sa
×</td><td>step
×</td><td>ext_trig
×</td><td>gray
×</td><td>delay
×</td><td colspan="3">mode
×</td></tr>
<tr><td>VSG reg. 1</td><td colspan="7"></td><td colspan="2">prediv
×</td></tr>
<tr><td>VSG reg. 2</td><td></td><td colspan="8">Len
×</td></tr>
<tr><td>VSG reg. 3</td><td></td><td colspan="8">Fen
×</td></tr>
<tr><td>VSG reg. 4</td><td colspan="9">x_base
MODE 0: ⇔(X) MODE 2,3: ×</td></tr>
<tr><td>VSG reg. 5</td><td></td><td colspan="8">y_base
⇔(Y)</td></tr>
<tr><td>VSG reg. 6</td><td colspan="9">x_offset
×</td></tr>
<tr><td>VSG reg. 7</td><td></td><td colspan="8">y_offset
×</td></tr>
<tr><td>VSG reg. 8</td><td colspan="6"></td><td colspan="3">pipedelay
×</td></tr>
<tr><td>VSG reg. 9</td><td></td><td>chaninv
×</td><td></td><td>feninv
×</td><td>leninv
×</td><td>clkinv
×</td><td>clkoff
×</td><td>pxlclk
×</td><td>adcclk
×</td></tr>
</table>

(×: value is not changed ⇔: value is changed according to parameter **0**: value = 0 etc.)

MODE m

• Determination of the readout mode

0x09	m

Length of the command: 2 byte

Return: 1 byte

Error code

Error code: [0] ok

[249] no symmetric frame area
A switching from single channel mode to dual channel mode is only possible when an **axial-symmetric** frame area is set!

[250] The CAMCLK frequency selected before is not random in the readout mode set.
(e.g. 1MHz selected, afterwards command MODE 3)

m: Mode (0,2,3)

The readout mode can be determined with the command MODE:

MODE 0: single channel mode
MODE 2: dual channel mode, both halfs of the sensor are convergingly read out
MODE 3: dual channel mode, both halfs of the sensor are divergently read out

The following parameter remain:

- Frequency f_{CAMCLK}
- Number of clocks LEN, FEN
- Polarity CAMCLK, LEN, FEN
- Frame size and position

Configuration:

Bit	8	7	6	5	4	3	2	1	0
HDRC reg.		lrm **1**	Resdir **1**	nores **1**	gray **1**	rowlm ⇔	Collm ⇔	rowrm ⇔	colrm ⇔
VSG reg. 0	lof ×	sa ×	Step **0**	ext_trig ×	gray **1**	delay **1**	mode ⇔		
VSG reg. 1								prediv ⇔	
VSG reg. 2		Len ⇔							
VSG reg. 3		Fen ×							
VSG reg. 4	x_base ⇔								
VSG reg. 5		y_base ⇔							
VSG reg. 6	x_offset ⇔								
VSG reg. 7		y_offset ×							
VSG reg. 8							pipedelay **3**		
VSG reg. 9		chaninv ×		feninv ×	leninv ×	clkinv ×	Clkoff ×	pxlclk ×	adcclk ×

(×: value is not changed ⇔: value is changed according to parameter **0**: value = 0 etc.)

The HDRC register consists of the following:

Readout mode	0	2	3
HDRC register	0xF0	0xF1	0xF4

LEN clocks, polarity

• Configuration of the LINE-ENABLE signal

0x0A	clocks	polarity

Length of the command: 3 byte

Return: 1 byte

Error code

clocks: Length of the LEN signal, number of CAMCLK clocks of the LEN signal during its inactive phase
MODE 0: (0, 1, 2,...254, 255)
MODE 1, 2, 3: (0, 2, 4,...252, 254), (LSB of „clocks“ is ignored)

polarity: 1: LEN H active
0: LEN L active

The LEN command determines length and polarity of the inactive phase of the LINE-ENABLE signal as multiple of the signal CAMCLK.

Configuration by the LEN command:

Bit	8	7	6	5	4	3	2	1	0
VSG reg. 0	lof ×	sa ×	Step ×	ext_trig ×	gray ×	delay ×	mode ×		
VSG reg. 1								prediv ×	
VSG reg. 2		Len ⇔ (see table)							
VSG reg. 3		Fen ×							
VSG reg. 4	x_base ×								
VSG reg. 5		y_base ×							
VSG reg. 6	x_offset ×								
VSG reg. 7		y_offset ×							
VSG reg. 8							pipedelay ×		
VSG reg. 9		chaninv ×		feninv ×	leninv ⇔	Clkinv ×	clkoff ×	pxlclk ×	adcclk ×

	VSG reg. 2 (LEN register)	Leninv
MODE 0	Clocks	polarity
MODE 1, 2, 3	Clocks/2	polarity

(×: value is not changed ⇔: value is changed according to parameter **0**: value = 0 etc.)

FEN **rows, polarity**

• Configuration of the FRAME-ENABLE signal

0x0B	rows	Polarity

Length of the command: 3 byte

Return: 1 byte

Error code

clocks: Length of the FEN signal, number of rows of the FEN signal during its inactive phase

rows=0:	FEN becomes inactive in addition to LEN for frame synchronization
zeilen>0:	FEN becomes inactive for *n* rows for frame synchronization

polarity:

1:	FEN H active
0:	FEN L active

The FEN command determines length and polarity of the FRAME-ENABLE signal during its inactive phase.

Configuration of the FEN command:

Bit	8	7	6	5	4	3	2	1	0
VSG reg. 0	lof ×	sa ×	Step ×	ext_trig ×	gray ×	delay ×	mode ×		
VSG reg. 1								prediv ×	
VSG reg. 2		len ×							
VSG reg. 3		fen ⇔ (see table)							
VSG reg. 4	x_base ×								
VSG reg. 5		y_base ×							
VSG reg. 6	x_offset ×								
VSG reg. 7		y_offset ×							
VSG reg. 8							pipedelay ×		
VSG reg. 9		chaninv ×		feninv ⇔	leninv ×	clkinv ×	clkoff ×	pxlclk ×	adcclk ×

(×: value is not changed ⇔: value is changed according to parameter **0**: value = 0 etc.)

CAMCLK freq, polarity

• Configuration of the CAMCLK signal

0x0C	freq	polarity

Length of the command: 3 byte

Return: 1 byte

Error code

freq: Frequency in MHz: 1 (Mode 0 only), 2, 4, 8, 16 (Mode 2, 3 only)

polarity:
1: datas valid for HL edge, data exchange LH edge
0: datas valid for LH edge, data exchange HL edge

The CAMCLK command determines frequency and polarity of the camera clock.

Configuration by the CAMCLK command:

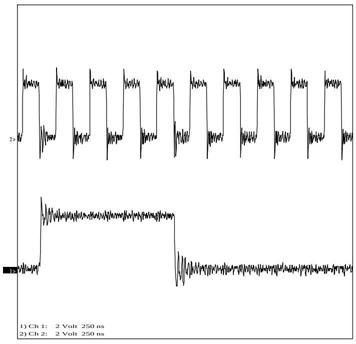
Bit	8	7	6	5	4	3	2	1	0
VSG reg. 0	lof ×	sa ×	step ×	Ext_trig ×	gray ×	delay ×	mode ×		
VSG reg. 1								prediv ⇔ (see table)	
VSG reg. 2		Len ×							
VSG reg. 3		Fen ×							
VSG reg. 4	x_base ×								
VSG reg. 5		y_base ×							
VSG reg. 6	x_offset ×								
VSG reg. 7		y_offset ×							
VSG reg. 8							Pipedelay ×		
VSG reg. 9		chaninv ×		feninv ×	leninv ×	clkinv ⇔	clkoff ×	pxlclk ×	Adcclk ×

Configuration of the PREDIV register (VSG reg. 1)

	1MHz	2MHz	4MHz	8MHz	16MHz
MODE 0	3	2	1	0	-
MODE 2, 3	-	3	2	1	0

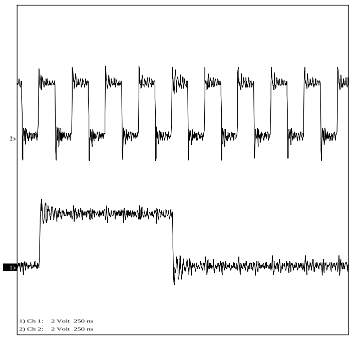
(×: value is not changed ⇔: value is changed according to parameter **0**: value = 0 etc.)

Configuration examples: CAMCLK, LEN, FEN signal



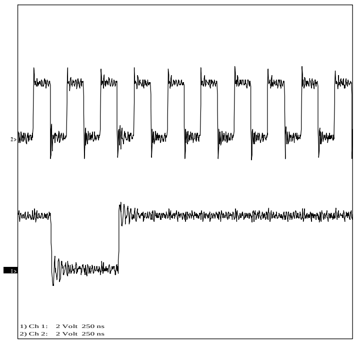
at the top: CAMCLK
at the bottom: LEN

Configuration: **CAMCLK 4, 0**
LEN 4, 0



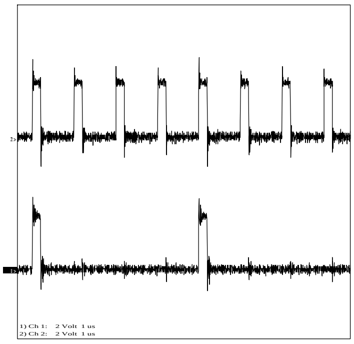
at the top: CAMCLK
at the bottom: LEN

Configuration: **CAMCLK 4, 1**
LEN 4, 0



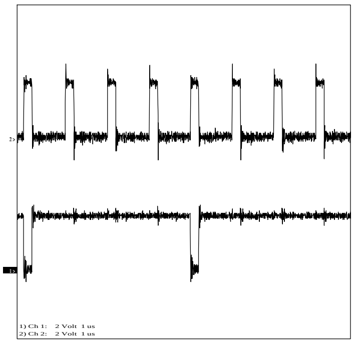
at the top: CAMCLK
at the bottom: LEN

Configuration: **CAMCLK 4, 0**
LEN 2, 1



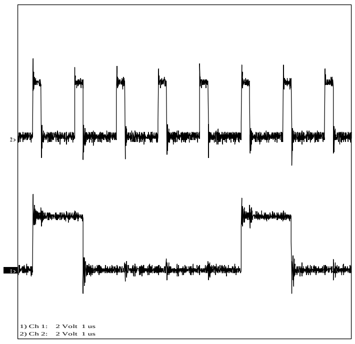
at the top: LEN at the bottom: FEN

Configuration: **FEN 0, 0**
Frame size (4×4)pixel



at the top: LEN at the bottom: FEN

Configuration: **FEN 0, 1**
Frame size (4×4)pixel



at the top: LEN at the bottom: FEN

Configuration: **FEN 1, 0**
Frame size (4×4)pixel

The polarity of the synchronuous signals which needs to be set depends on the type of the attached framegrabber.

Example:

Manufacturer	Type	CAMCLK, LEN, FEN parameter
ELTEC, Mainz	PCEye, CAMA 160	CAMCLK n_1, 0 LEN n_2, 0 FEN n_3, 0
INO-Vision	INOCAP CSD	CAMCLK n_1, 0 LEN n_2, 1 FEN n_3, 1

ROT

• Rotation of the frame about 180°

0x0D

Length of the command: 1 byte

Return: 1 byte

Error code

Configuration of the ROT command:

Bit	8	7	6	5	4	3	2	1	0
HDRC reg.		lrm ⇔	Resdir ×	nores ×	gray ×	rowlm ⇔	Collm ⇔	rowrm ⇔	colrm ⇔

MIR

• Mirroring of both sensor halfs

0x0E

Length of the command: 1 byte

Return: 1 Byte

Error code

Configuration of the MIR command:

Bit	8	7	6	5	4	3	2	1	0
HDRC reg.		lrm ⇔	Resdir ×	nores ×	gray ×	rowlm ×	collm ⇔	rowrm ×	colrm ⇔

EEPROM

• Readout of the internal configuration eeprom

0x0F

Length of the command: 1 byte

Return: 130 byte

0x0F	Contents Adr. 0x00	Contents Adr. 0x01	Contents Adr. 0x02	Contents Adr. 0x03			Contents Adr. 0x7F	Error code

The EEPROM command outputs the contents of the configuration EEPROM over the configuration interface. It is possible to read out all configuration parameters with this. The interpretation of the datas takes place on the controlling PC.

Data assignment in the internal EEPROM:

Address	Byte	Function
0x00	1	HDRC register
0x01	1	Multiplexer register
0x02-0x15	10×2	VSG register 0...9
0x16-0x19	4	DAC channels 0...3
0x1A-0x29	4×4	DAC backup for 4 correction tables
0x2A	1	Ones complement of the less valuable bytes of the total over the address 0x1A-0x29 (proof total)
0x2B	1	Set correction table
0x2C	8	Interface specific datas
0x34	2	BRIGHT reference
0x36	2	DARK reference
0x38	1	GAIN difference
0x39	1	OFFSET difference
0x3A	10	Statistical data correction table0
0x44	10	Statistical data correction table1
0x4E	10	Statistical data correction table2
0x58	10	Statistical data correction table3

INTERFACE par1, par2

• Setting of the interface-specific parameter

0x10	par1	high (par1)	low (par1)

Length of the command: 4 byte

Return: 1 byte

Error code

Error code: [0] ok

The modified parameters of the configuration interface can be determined with the INTERFACE command.

The meaning of parameter *par1* and *par2* depends on the used configuration interface (standard RS232C).

Configuration interface RS232C

The camera permits the determination of a user-specific interface configuration each for the plain text and the HEX configuration (rate of transmission, number of stop and start bits, parity bits). By setting the configuration switch (setting 13 for plain text configuration, setting 14 for HEX configuration) on the camera back and pressing the RESET button afterwards, the setting selected is taken over.

The user-specific interface configuration can be undone by shifting the configuration switch to setting 0...12 at any time.

Meaning of the Parameter :

par1 **=0:** Setting of interface parameter for plain text configuration
=1: Setting of interface parameter for HEX configuration

par2: Interface parameter word

Structure:

Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	=0: 8bit =1: 7bit	Parity: =00: none =10: EVEN =11: ODD		Stopbit: =0: 1 =1: 2	0	Predivider n		Transmission rate N							

The Baud rate ***B*** results from parametern **n** and **N** as follows:

$$N = \frac{\Phi}{2^{(2n+5)} \cdot B} - 1 \qquad B = \frac{\Phi}{2^{(2n+5)} \cdot (N+1)} \qquad \Phi = 8 \cdot 10^6\,Hz$$

Error ***E*** of the Baud fate in %:

$$E = \left(\frac{\Phi}{(N+1) \cdot B \cdot 2^{(2n+5)}} - 1 \right) \cdot 100\% \qquad \Phi = 8 \cdot 10^6 \, Hz$$

Example:

Baud rate [Bd]	n	N	Error [%]
110	2	141	0,03
150	2	103	0,16
300	1	207	0,16
600	1	103	0,16
1200	0	207	0,16
2400	0	103	0,16
4800	0	51	0,16
9600	0	25	0,16
19200	0	12	0,16
31250	0	7	0

Interface parameter for plain text configuration: 4800Bd, 2 stop bits, parity EVEN:

INTERFACE 0, 0x2833
INTERFACE 0, 10291

Setting of a user-specific transmission rate:

1. Obeying the INTERFACE command in plain text or HEX mode
2. Setting of the configuration switch (setting 13: plain text mode, setting 14: HEX mode)
3. Pressing the RESET button or restarting the camera by switching off and on the power supply.

Notes: In HEX mode and when transmitting datas with the XMODEM control protocol in plain text mode, always 8 data bits need to be transmitted.

TAB table

• Setting of a correction table

0x11	table

Length of the command: 2 byte

Return: 1 byte

Error code

The TAB command selects a correction table. The following assignment is valid:

<table>
<tr><td>TAB</td><td colspan="3"></td></tr>
<tr><td>0</td><td>Correction table 0 from FLASH-ROM</td><td rowspan="6"></td><td rowspan="12"></td></tr>
<tr><td>1</td><td>Correction table 1 from FLASH-ROM</td></tr>
<tr><td>2</td><td>Correction table 2 from FLASH-ROM</td></tr>
<tr><td>3</td><td>Correction table 3 from FLASH-ROM</td></tr>
<tr><td>4</td><td>Not corrected frame</td></tr>
<tr><td>5</td><td>Test pattern for GAIN and OFFSET</td></tr>
<tr><td>6</td><td>Correction table 0 from FLASH-ROM</td><td rowspan="6">Additional inlaying of the main setting datas
• X direction: GAIN
• Y direction: OFFSET</td></tr>
<tr><td>7</td><td>Correction table 1 from FLASH-ROM</td></tr>
<tr><td>8</td><td>Correction table 2 from FLASH-ROM</td></tr>
<tr><td>9</td><td>Correction table 3 from FLASH-ROM</td></tr>
<tr><td>10</td><td>Not corrected frame</td></tr>
<tr><td>11</td><td>Test pattern for GAIN and OFFSET</td></tr>
<tr><td>12</td><td>Correction table 0 from FLASH-ROM</td><td rowspan="6"></td><td rowspan="12">Additional inlaying of the frame size</td></tr>
<tr><td>13</td><td>Correction table 1 from FLASH-ROM</td></tr>
<tr><td>14</td><td>Correction table 2 from FLASH-ROM</td></tr>
<tr><td>15</td><td>Correction table 3 from FLASH-ROM</td></tr>
<tr><td>16</td><td>Not corrected frame</td></tr>
<tr><td>17</td><td>Test pattern for GAIN and OFFSET</td></tr>
<tr><td>18</td><td>Correction table 0 from FLASH-ROM</td><td rowspan="6">Additonal inlaying of the main setting datas
• X direction: GAIN
• Y direction: OFFSET</td></tr>
<tr><td>19</td><td>Correction table 1 from FLASH-ROM</td></tr>
<tr><td>20</td><td>Correction table 2 from FLASH-ROM</td></tr>
<tr><td>21</td><td>Correction table 3 from FLASH-ROM</td></tr>
<tr><td>22</td><td>Not corrected frame</td></tr>
<tr><td>23</td><td>Test pattern for GAIN and OFFSET</td></tr>
</table>

HIGH **parameter** (privileged command, see $ command)

• Definition of the BRIGHT reference (see CAL command)

0x12	high (parameter)	low (parameter)

Length of the command: 3 byte

Return: 1 byte

Error code

LOW **parameter** (privileged command, see $ command)

• Definition of the DARK refernce (see CAL command)

0x13	high (parameter)	low (parameter)

Length of the command: 3 byte

Return: 1 byte

Error code

CAL **table, parameter** (privileged command, see $ command)

• Calibration of the camera

0x14	table	parameter

Length of the command: 3 byte

table: No. of the correction table in FLASH-ROM (aim for calibration datas, 0...3)

parameter:
0: rough correction only, correction table is described as mean value
1: OFFSET fine correction
2: GAIN fine correction
3: GAIN and OFFSET fine correction

Return: 1 byte

Error code

The CAL command serves the automatic trimming of the camera on a calibrated light generator. At the same time the camera parameter are varied this way that a defined luminance results in a certain absolute value and a luminance jump leads to a defined value difference on the exit.

The numerical values for the high and low luminance are reconcilled with HIGH or LOW. The following connection exists between luminance and numerical value.

$$\frac{L}{L_0} = 683\frac{1}{t \cdot p} \cdot \left(\frac{k}{k_0}\right)^2 \cdot 2^{\left(\frac{Z}{32}-16\right)}$$

Z =numerical value(10bit)

L: luminance of the light generator in cd/m^2
L_0=4cd/m^2

k: aperture of the lens
k_0=1

t : Transmission degree of the lens (approx. 0,8)

The setting of different luminances is required by the blinking of the camera LED:

yellow-red: high luminance
yellow-green: low luminance
yellow: calibration is in progress

The change of luminance has to be acknowledged by pressing the button on the camera back (button next to RESET). The use of an automatically controlled light generator is described in chapter „Operation with automatic light generator“.

WR adr, date

(privileged command, see $ command)

• Description of the internal EEPROMS

0x15	adr	date

Length of the command: 3 byte

Return: 1 byte

Error code

Error code: [0] ok

adr: EEPROM address (0...127)
date: 0...255

Attention: This command serves test patterns only! An overwriting of the internal proof totals by accident results in an initialization of the camera connected with losing all calibration data.

TRIG par

• Switching on of the external frame trigger

0x16	par

Length of the command: 2 byte

Return: 1 byte

Error code

Error code: [0] ok

par =0: External frame trigger OFF
The readout of the sensor is continuously done

par =1: External frame trigger ON
With a rising edge on the FRAME_TRIG input one frame is read out. Afterwards the camera remains in waiting state.

Configuration:

Bit	8	7	6	5	4	3	2	1	0
VSG reg. 0	lof ×	sa ×	step ×	Ext_trig ⇔	gray ×	delay ×	mode ×		

(×: value is not changed ⇔: value is changed according to parameter **0**: value = 0 etc.)

ADC channel

(privileged command, see $ command)

• Enquiring the internal AD converter for test purposes

0x17	channel

Length of the command: 2 byte

Return: 4 byte

0x17	high (adc_parameter)	low (adc_parameter)	Error code

Error code: [0] ok

adc_parameter: value of voltage in mV

// (Comment)

• Marking of an command line as comment

0x18

Length of the command: 1 byte

Return: 1 byte

Error code

Error code: [0] ok

The „//“ command marks a command line as comment and improves the overall view of configuration files in connection with the TERMINAL menu point „Send text file“. The camera settings are not changed during the obeying process.

The application is only useful in plain text mode.

Example:

Text file INO_VIS.TXT for configuring the camera for a connection to an INO-VISION framegrabber, created with WRITE (Windows) or EDIT (DOS):

```
// INO-Vision configuration, 8MHz
// LINE-, FRAME-Enable LOW active
len 16,1
fen 0,1
camclk 8,0
```

GAIN gain_difference

• Setting of a gain difference

0x19	gain_difference

Length of the command: 2 byte [1]

Return: 1 byte

Error code

Error code: [0] ok
[253] illegal parameter
gain_difference: parameter for gain difference, range 0...45

The „GAIN" command permits to increase the gain of the video amplifier. At the same time a constant difference to the value of the DAC register which is determined when calibrating the camera is laid down. This results in a constant rise of the gain independent of the calibration results. The set gain difference is considered as independent of the correction table set.

The connection of the parameter „gain_differenz" *DZ* and the rise of the gain results from the following equation:

$$\Delta V = \Delta Z \cdot 0{,}136 dB$$

Example:

ΔZ	15	22	29	37	44
ΔV in *dB*	≈2	≈3	≈4	≈5	≈6

OFFSET offset

• Setting of the brightness offset

0x1A	offset

Length of the command: 2 bytes

Return: 1 byte

Error code

Error code: [0] ok
[253] illegal parameter

offset: parameter of the brightness offset, range 0...50

The „Offset" command permits a change of the frame brightness independent of the DAC offset parameter determined when trimming the camera. Consequently, the ascertainable brightness range can be determined in connection with a gain rise by the gain command. A higher value for „offset" results in a darker frame. The set offset parameter is valid for all correction tables.

STAT table (privileged command, see $ command)

• Enquiring statistical parameter of the correction table „table“ for control purposes

0x1B	table

Lenght of the command: 2 byte

Return: 32 byte

0x1B	Byte 2	...	Byte 31	fehler-code

Error code: [0] ok
[253] unvalid parameter

table: correction table 0...3

With the STAT command it is possible to make a statement about the statisitical distribution of the corres-ponding correction files. Byte 2...31 contain the following information:

Byte #		Meaning		
High	Low			
2	3	Sensor data	**Bit 0,1:** selected parameter of the „CAL“ command 0: rough cal. 1: OFFSET 2: GAIN 3: GAIN+OFFSET	**Bit 2...15:** Difference INSTANTANEOUS-DESIRED value of the mean brightness figures after the calibration (complement on two)
4	5		Standard drift (68% interval), sensor roughly corrected	
6	7		99% interval, sensor roughly corrected	
8	9		Standard drift (68% interval), sensor fine corrected	
10	11		99% interval, sensor fine corrected	
12	13	Offset-correction file	Mean value instantaneous value (desired value: 512)	
14	15		68% interval (standard drift), =0 when correction file is empty	
16	17		99% interval	
18	19		90% interval	
20	21		50% interval	
22	23	Gain-correction file	Mean value instantaneous value (desired value: 32)	
24	25		68% interval (standard drift)	
26	27		99% interval	
28	29		90% interval	
30	31		50% interval	

4.3 Commands for transmitting bigger data quantities

The internal software of the *LOGLUX*® camera permits a transmission of bigger data quantities with the XMODEM control protocol. The protocol provides an error control of the data transmitted. The command for the XMODEM data transmission

1. *LOAD par1, par2*
2. *SAVE pars*

are available for certain camera setting only. As the XMODEM transmission relies on 8 transmitted data bits, the following settings need to be adjusted:

1. configuration in plain text mode
2. quantity of transmitted data bits of the RS232 interface: 8bit

LOAD par1, par2

- Transmitting a file from control computer to camera

par1: Determination of the contents of the file which is to transmit

0: correction table 0
1: correction table 1
2: correction table 2
3: correction table 3

par2: Determination of the control protocol

0: Requirements XMODEM control protocol
1: Requirements XMODEM/CRC control protocol

4.3.1 Sending of files in XMODEM format with the WIN3.1x -, WIN 95- TERMINAL (VT100 emulation)

Windows 3.1x-Terminal:

1. Menu „setting“ (only when configuring the terminal for the first time)

   1.1 Menu „binary transmission“
   - Activate **XMODEM/CRC**

   1.2 Menu „modem commands“
   - Activate **NONE**

2. Input camera command „LOAD par, 1↵“
3. Wait for „LOGLUX ready for receiving a binary file...“
4. Menu „transmission“
   4.1 Menu „SEND BINARY FILE“
   4.2 Select file
   4.3 OK

Windows 95 - HyperTerminal:

1. Send camera command „LOAD *par1,par2*↵“
2. Wait for „ LOGLUX ready for receiving a binary file...“
3. Menu „transmission“
   3.1 Menu „SEND FILE“
   3.2 Select file
   3.3 Activate **Xmodem**
   3.4 OK

Should any problems arise when transmitting datas (transmission does not start), select menu „FILE →PREDICATES →SELECTION NUMBER →CONFIGURATION →PROTOCOL“ **„NONE“**.

SAVE par

- Transmitting a file from camera to control computer

par: Determining the contents of the file which is to transmit

par	File	
0	**Binary file:** correction table 0	
1	**Binary file:** correction table 1	
2	**Binary file:** correction table 2	
3	**Binary file:** correction table 3	
4	**Windows *.BMP file**	Bit 2...9 of the recorded 10bit frame
5	Palett: gray scales	Restricted value range 1 (see below)
6		Restricted value range 2 (see below)
7	**Windows *.BMP file**	Bit 2...9 of the recorded 10bit frame
8	Palett: gray scales	Restricted value range 1 (see below)
9	Marking overshoot of range	Restricted value range 2 (see below)
10	**Binary file:** single frame 10bit	
11	**Text file*.TXT**	Set camera parameter
12	ASCII code	Calibration statistics
13		Set camera parameter, configuration statistics
14		Calibration statistics for EXCEL table
15		10bit single frame histogramm for EXCEL
16		Histogramm offset correction table 0 for EXCEL
17		Histogramm gain correction table 0 for EXCEL
18		Histogramm offset correction table 1 for EXCEL
19		Histogramm gain correction table 1 for EXCEL
20		Histogramm offset correction table 2 for EXCEL
21		Histogramm gain correction table 2 for EXCEL
22		Histogramm offset correction table 3 for EXCEL
23		Histogramm gain correction table 3 for EXCEL

Calculation of the restricted value range:

Processing steps:
1. Calculation of the mean frame brightness *M*
2. Subtraction of the mean frame brightness from all 10bit pixel values *P(x,y)* (complement on two)
3. Assigning the difference *D(x,y)=P(x,y)-M* according to the following scheme:

Range 1: ***if*** *D(x,y) < -256*
then *D(x,y) = -256*

if *D(x,y) > 255*
then *D(x,y) = 255*

gray scale(x,y) = D(x,y)/2+128

range 2: ***if*** *D(x,y) < -128*
then *D(x,y) = -128*

if *D(x,y) > 127*
then *D(x,y) = 127*

gray scale(x,y) = D(x,y)+128

4.3.2 Receiving of files in XMODEM format with the WIN3.1x -, WIN 95-TERMINAL (VT100 emulation)

Windows 3.1x-Terminal:
1. Menu „settings“ (only when configuring the terminal for the first time)

   1.1 Menu „Binary transmission“
   - Activate **XMODEM/CRC**

   1.2 Menu „Modem commands“
   - Activate **NONE**

2. Input camera command „SAVE par↵“
3. Wait for „ LOGLUX ready for sending a binary file...“
4. Menu „Transmission“
   4.1 Menu „BINARY FILE RECEIVED“
   4.2 Select file
   4.3 OK

Windows 95 - HyperTerminal:
1. Send camera command „LOAD *par1,par2*↵“
2. Wait for „ LOGLUX ready for sending a binary file...“
3. Menu „Transmission“
   3.1 Menu „Send file“
   3.2 Select file which is to receive
   3.3 activate **Xmodem**
   3.4 OK

Should any problems arise when transmitting datas (transmission does not start), select menu „FILE →PREDICATES →SELECTION NUMBER →CONFIGURATION →PROTOCOLL“ „NONE“.

4.3.3 Structure of the correction file

One correction file is sent by the camera when obeying the command *SAVE 0...3* and has the following structure:

Lenght of file, total:		262170 bytes
Consisting of:	file head:	26 bytes
	correction data	262144 bytes (0,25 MB)

Structure of the data head:

Byte		Description	
0...7		Character string: „LOGLUX“	
8		Device code: =0	
9		Software version:	year
10			month
11			day
12		Calibration values DAC register	GAIN, reg. 0
13			OFFSET, reg. 1
14			ADCREF, reg. 2
15			VRESET, reg. 3
16	17	Statistical data, Sensor calibration	Drift of desired value
18	19		68% interv., uncorr.
20	21		99% interv., uncorr.
22	23		68% interv., corr.
24	25		99% interv., corr.

Structure of correction data block:

Byte		Structure 16bit word	Correction data: pixel coordinate		
HIGH	LOW			X	Y
0	1	Bit 0...9: OFFSET Bit 10...15: GAIN	Left sensor half	0	0
...	...			...	...
510	511			0	255
512	513			1	0
...	..			...	...
131070	131071			255	255
131072	131073		Right sensor half	256	0
...	...			...	...
131582	131583			511	0
131584	131585			256	1
...	...			...	...
262142	262143			511	255

Please note: The coordinate origin is located in the upper left-hand corner of the sensor (NW corner).

[0,0]	...	[255,0]	[256,0]	...	[511,255]
...	...	...	...	...	...
[0,255]	...	[255,255]	[256,255]	...	[511,255]

4.3.4 Structure of a 10bit frame file

Beginning with the coordinate [0,0] pixel values are ascendingly saved as 16bit words in a 10bit frame file. Bit 10...15 is set =0. This results in file size of 0,25Mbyte = 256kByte. The higher-quality byte of the 16bit word is located on a lower address:

Address n HIGH-Byte	Address (n+1) LOW-Byte

Structure of the 10bit frame file:

[0,0] 0x00000	...	[255,0] 0x001FE	[256,0] 0x00200	...	[511,255] 0x003FE
...	...	...	...	...	...
[0,255] 0x3FC00	...	[255,255] 0x3FDFE	[256,255] 0x3FE00	...	[511,255] 0x3FFFE

4.3.5 Structure of a text file for infixing a MS-EXCEL table

The text files which are to infix in Microsoft EXCEL have the following characteristics:

1. Characters of the ASCII code are transmitted only (<0x80)
2. **Data columns** are seperated by **commas**.
3. **Data rows** are seperated by **CR** (0x0D) **LF**(0x0A).
4. **Numbers** are given in decimal style.
5. **Character strings** are marked by quotation marks (**"**).

Example: A text file with the following contents (part) is interpreted by MS-EXCEL as shown below:

Text file   *.txt

```
"LOGLUX histogramm(GAIN)",0,0
,1,0
,2,0
,3,0
,4,0
,5,0
,6,0
...
```

EXCEL table   *.xls

LOGLUX histogramm(GAIN)	0	0
	1	0
	2	0
	3	0
	4	0
	5	0
	6	0

4.3.6 Description of the XMODEM control protocol

4.3.6.1 General features

The XMODEM control protocol permits a transmission of bigger data quantities over a serial interface. The transmission is carried out in 128byte blocks. Should a transmission error arise, the uncorrect block is transmitted once more. A further development of the XMODEM control protocol is the XMODEM/CRC control protocol. It is more fault-tolerant because of the use of a CRC checksum (CRC: Cyclical Redundancy Check). A transmission with the XMODEM control protocol requires the possibility to transmit 8 data bits. This must be done without a control with the XON/XOFF control protocol.

In principel every data block is structured as follows:

ASCII characters SOH **0x01**	Block number	(255-Block no.)	128 Data bytes	checksum XMODEM: 1byte XMODEM/CRC: 2byte

XMODEM and XMODEM/CRC data blocks differ from each other only in their structure of the checksum.

4.3.6.2 XMODEM control protocol

Data transmission with the XMODEM control protocol

Data sender	Data receiver
	1. Receiver signals standby with sending the ASCII character NAK (0x15)
2. (Cancel of the data transmission by sending the ASCII character CAN (0x18), then immediate stop of transmission)	
3. Sending of a data block (132 byte)	
	4. (Cancel of the data transmission by sending the ASCII character CAN (0x18), then immediate stop of transmission)
	5. Controlling the checksum
	6. Sending the ASCII character ACK (0x06) when the data transmission is ok, otherwise NAK (0x15)
7. (Cancel of the data transmission by sending the ASCII character CAN (0x18), then immediate stop of transmission)	
8. When receiving NAK the data block is sent once more, otherwise sent next data block	
9. Repeat steps 4...8 until all data blocks are transmitted	
10. Send the ASCII character EOT (0x04) for stoping the data transmission	
	11. Sending the ASCII character NAK (0x15)
12. Sending the ASCII character EOT (0x04)	
	13. Sending the ASCII character ACK (0x06)
14. Transmission finished	

Calculation of the checksum for the XMODEM control protocol (1byte):

checksum = (Sum over all 128 data bytes) & 255

4.3.6.3 XMODEM/CRC control protocol

Data transmission with the XMODEM/CRC control protocol

Data sender	Data receiver
	1. Receiver signals standby with sending the ASCII character „C“ (0x43)
2. (Cancel of the data transmission by sending the ASCII character CAN (0x18), then immediate stop of transmission)	
3. Sending a data block (133 bytes)	
	4. (Cancel of the data transmission by sending the ASCII character CAN (0x18), then immediate stop of transmission)
	5. Controlling the checksum
	6. Send the ASCII character ACK (0x06) when the data transmission is ok, otherwise NAK (0x15)
7. (Cancel of the data transmission by sending the ASCII character CAN (0x18), then immediate stop of transmission)	
8. After receiving NAK, send the data block once more, otherwise send next data block	
9. Repeat steps 4...8 until all data blocks are transmitted	
10. Sending the ASCII character EOT (0x04) for stopping the data transmission	
	11. Sending the ASCII character NAK (0x15)
12. Sending the ASCII character EOT (0x04)	
	13. Sending the ASCII character ACK (0x06)
14. Transmission is finished	

Calculation of the checksum for the XMODEM/CRC control protocol (2byte), 1 Data block:

```
crc = 0

FOR z=0 TO 127
        crc = (crc*256) xor (crctabelle[(crc/256) xor data byte[z]])
```

The Array „*crctabelle[...]*“ is a constant. Its contents can be calculated when following these instructions:

```
FOR i=0 TO 255
        a = 0
        k=i*256

        FOR j=0 TO 7
                IF (((k xor a) and 0x8000) <>0)
                        a = (a*2) xor 4129

                ELSE
                        a = (a*2)

                k = k*2

        crctabelle[i] = a
```

The checksum is transmitted in the order HIGH byte, LOW byte.

4.4 Operation with automatic light generator

The control software of the `LOGLUX`Ò camera permits the attachment of an automatic light generator to the camera-internal I^{2}C bus. A 5-pole *JST* plug (CON6) which is on the circuit 6000 1 0400 (H8 board) from index C on is proveded for this and has the following load:

PIN no.	load
1	SCL (clock I^{2}C bus)
2	SDA (data signal I^{2}C bus)
3	GND
4	free
5	DV3,3

The initialization routine looks on the I^{2}C bus for connected subscriber with the addresses 0x40, 0x42, 0x44, 0x70, 0x72 and 0x74. If a subscriber is found in one of these addresses, the following message appears in plain text mode:

`Light generator addr.:` *`address`*

The calibration routine (command CAL) takes on an 8bit port on this address. Consequently, the use of the *Philips* I^{2}C port *PCF8574* or *PCF8574A* is possible.

The single bits of the IO port are described, selected or enquired as shown below:

Bit no.	Input/output	Meaning
0	Output	Aperture motor reverse action, LOW active
1	Output	Aperture motor right action, LOW active
2	Input	End position switch 1, LOW active (=LOW: end position 1 detected)
3	Input	End position switch 2, LOW active (=LOW: end position 2 detected)
4	Output	„BRIGHT“, LOW active
5	Output	„DARK“, LOW active
6	Output	„CALIBRATIO IS RUNNING“, LOW acktive
7	Output	„CALIBRATION IS FINISHED“, LOW active

According to the ascertained I^{2}C address the signals „reverse/right action“ are selected in different ways and the signals „end position 1/-2“ are interpreted in different ways.

<table>
<tr><th colspan="2">I^{2}C address</th><th colspan="2">Selecting scheme</th></tr>
<tr><th>PCF8574</th><th>PCF8574A</th><th>End position switch 1,2</th><th>Aperture motor outputs</th></tr>
<tr><td>0x40</td><td>0x70</td><td rowspan="2">End position switch 1 =LOW: „BRIGHT“
End position switch 2 =LOW: „DARK“</td><td>„BRIGHT“: reverse action =LOW
„DARK“: right action =LOW</td></tr>
<tr><td>0x42</td><td>0x72</td><td rowspan="2">„BRIGHT“ / „DARK“:
reverse action =LOW
right action: without meaning</td></tr>
<tr><td>0x44</td><td>0x74</td><td>End position switch 1 =LOW: „BRIGHT“
End position switch 1 =HIGH: „DARK“</td></tr>
</table>

When the camera trimming is finished, bit 7 („CALIBRATION FINISHED“) is activated for approx. 1s (acoustic signal).

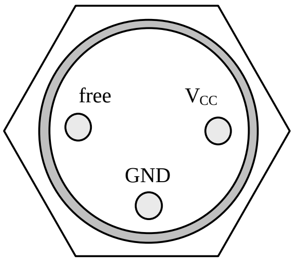
A determination of bit 1,2 of the I^{2}C address can be carried out over a DIP switch.

5 Glossary

6 Appendix

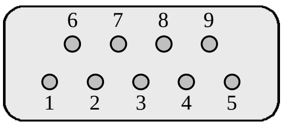
6.1 Table: Plug load

6.1.1 Power supply



Frame data Interface	V_{CC}	I_B	*Plug*
LVDS	5,5...6,2 V	450mA	*Binder* 09-0407-00-03
RS422		700mA	
LWL		650mA	*Thomas & Betts* T01-0560-P03

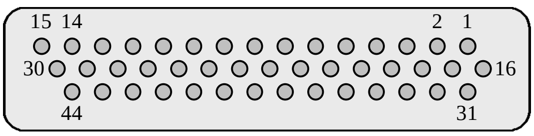
6.1.2 RS232/RS422 Configuration interface



9-pole D-Sub, plug

Pin no.	RS232	RS422 (optinal)
1	With 4,6 connected	
2	RxD	RxD+
3	TxD	TxD+
4	With 1,6 connected	
5	GND	
6	With 4,1 connected	
7	/RTS	RxD-
8	/CTS	TxD-
9	frei	

6.1.3 RS422/LVDS Frame data interface



1	D0+	16	D0-	31	SDA
2	D1+	17	D1-	32	SCL
3	D2+	18	D2-	33	GND
4	D3+	19	D3-	34	GND
5	D4+	20	D4-	35	GND
6	D5+	21	D5-	36	GND
7	D6+	22	D6-	37	frei
8	D7+	23	D7-	38	frei
9	D8+	24	D8-	39	frei
10	D9+	25	D9-	40	frei
11	LEN+	26	LEN-	41	3,3V
12	FEN+	27	FEN-	42	3,3V
13	CAMCLK+	28	CAMCLK-	43	5V
14	OTR+	29	OTR-	44	5V
15	FRAMETR+	30	FRAMETR-		

6.2 Table: Lenses for *LOGLUX®*

Designation	Focal length f' [mm]	Aperture range	Camera connection	Pict. angle (2σ')	Recording range	Distance setting
LOGLAR T 2/13	13,2	2 (fixed)	over 0 plate (M16 x 0,5)	49°	∞ ... 2m ("Fixed focus")	On customer's demand Fixed settings up to approx. 0,2m possible
DOCTAR 5/13,5	13,51	5 (fixed)	C-Mount plate (1" x 32 TPI)	48°	∞ ... 0,2m	Over lens connection thread
LOGLAR T 2,8/16	16,3	2,8 (fixed)	over 0 plate (M16 x 0,5)	41°	∞ ... 2,3m ("Fixed focus")	On customer's demand Fixed settings up to approx. 0,3m possible
LOGLAR T 4/16	16,3	4 (fixed)	over 0 plate (M16 x 0,5)	41°	∞ ... 1,6m ("Fixed focus")	On customer's demand Fixed settings up to approx. 0,3m possible
LOGLAR T 8/16	16,3	8 (fixed)	over 0 plate (M16 x 0,5)	41°	∞ ... 0,8m ("Fixed focus")	On customer's demand Fixed settings up to approx. 0,3m possible
TEVIDON 1,4/25	25,4	1,4 ... 16	C-Mount plate (1" x 32 TPI)	27°	∞ ... 0,3m	over lens setting ring
ROTAR T 4,5/29	29,85	4,5 ... 16	C-Mount plate (1" x 32 TPI)	23°	∞ ... 0,3m	Continuously up to approx. 5mm Lens extension over mount
Tessar T 3,5/37,5	37,5	3,5 (fixed)	C-Mount plate (1" x 32 TPI)	19°	∞ ... 0,2m	Continuously up to approx. 12mm Lens extension over mount
MAKRO-TV 4/45	45,3	4 (fixed)	C-Mount plate (1" x 32 TPI)	Up to 10°	79 ... 124mm Operation distance	over lens setting ring for macro range 1:1 to 1:2
ROTAR T 4,5/50 S	50,75	4,5 ... 22	C-Mount plate (1" x 32 TPI)	14°	∞ ... 0,4m	Continuously up to approx. 10mm Lens extension over mount
Tessar T 3,5/50	51,43	3,5 (fixed)	C-Mount plate (1" x 32 TPI)	14°	∞ ... 0,3m	Continouosly up to approx. 14mm Lens extension over mount
Tessar T 3,5/70	70,3	3,5 (fixed)	C-Mount plate (1" x 32 TPI)	10°	∞ ... 0,65m	Continuously up to approx. 10mm Lens extension over mount
ROTAR T 6,5/75	73,5	6,5 ... 22	C-Mount plate (1" x 32 TPI)	10°	∞ ... 0,3m	Continuously up to approx. 11mm Lens extension over mount
TEVIDON 2,8/100	96,5	2,8 ... 16	C-Mount plate (1" x 32 TPI)	8°	∞ ... 1,5m	over lens setting ring
LOGLAR 19/320 makro	317,3	19 (fixed)	C-Mount plate (1" x 32 TPI)	bis 2°	0,6 ... 3m or ∞ operation distance	Over adaptor for macro range 1:1 to 1:6 as well as for ∞

6.3 Optical supplementary components

6.3.1 Stereo attachment

Designation	Stereoscopic basis	stereoscopic distance range	
		Close point width	Far point width
Stereo attachment, small	12mm	0,5m	2,3m
Stereo attachment, big	60mm	3,0m	∞

6.3.2 Other

Designation	Description
C-Mount lens plate	For using C-Mount lenses
Microscope adaptor, mono	Suitable for Zeiss Jena microscopes with mf-adjustment
Microscope adaptor, stereo	
Microscope tubus	For microscope lenses according to DIN 58887 in standard connection thread

6.4 Accessories

6.4.1 Mechanic accessories

Designation	Description
Fastening block	
Tripod attachment	Tripod thread A 1/4″ and A 3/8″ according to DIN 4503/1
Notch	Thread ∅M5

6.4.2 Accessories for the frame data collection, power supply and camera control

Designation	Description
PCI-Frame Grabber INOCAP INO VISION	Digital framegrabber, PCI bus
PCI-Frame Grabber HC-32 PCI HaSoTec	
Frame data cable	Feeder cable, framegrabber camera for LVDS or RS422-frame data interface
Zero modem cable	Feeder cable, camera configuration interface – COM port PC
Power supply cable	Power supply plug 3-pole, conducter end bushings
Power supply	6V/1A, power supply plug 3-pole

6.5 Outlines: lenses and accessories

1. LOGLAR 19/320
2. Microscope tubus
3. Microscope adaptor, mono
4. Microscope adaptor, stereo
5. Stereo adaptor, small
6. Stereo adaptor, big
7. Fastening block